DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

EMULATION OF THE AN/UYK-20
TACTICAL DATA COMPUTER
ON THE BURROUGHS D-MACHINE

by

Ralph Harry Anzelmo
and
Theodore Lawrence Kaye

March 1977

Thesis Advisor: L. V. Rich

Approved for public release; distribution unlimited.

T 177953
[redacted]

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

Emulation of the AN/UYK-20 Tactical Data Computer on the Burroughs D-machine

5. TYPE OF REPORT & PERIOD COVERED

Master's Thesis; March 1977

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Ralph Harry Anzelmo and
Theodore Lawrence Kaye

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

March 1977

13. NUMBER OF PAGES

263

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

Naval Postgraduate School
Monterey, California 93940

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

emulation
microprogramming
AN/UYK-20
Burroughs D-machine

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A representation of the Univac AN/UYK-20 computer systems has been emulated on the Burroughs D Interpreter-Based System. The entire AN/UYK-20 instruction repertoire has been emulated with the exception of the "math pac" option (floating point arithmetic and cordic functions), clock and interrupt codes, and input/output operation codes. Modular design with extensive documentation has been implemented throughout program

(cont.)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

(Page 1)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (cont.)

development allowing for ease of modification and further extensions to the existing emulation. Emulation and hardware architecture of the AN/UYK-20 and the Burroughs D-machine, are discussed in conjunction with the AN/UYK-20 itself. Methods of testing and debugging, sample test programs and recommendations for continued design modifications to the emulation are presented.

Approved for public release; distribution unlimited.

Emulation of the AN/UYK-20 Tactical Data Computer on the Burroughs D-machine

by

Ralph Harry Anzelmo
Captain, United States Marine Corps
B.A., Montclair State College, 1968

Theodore Lawrence Kaye
Lieutenant, United States Navy
B.S.S.E., United States Naval Academy, 1972

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
MARCH 1977

ABSTRACT

A representation of the Univac AN/UYK-20 computer system has been emulated on the Burroughs D Interpreter-Based System. The entire AN/UYK-20 instruction repertoire has been emulated with the exception of the 'math pac' option (floating point arithmetic and cordic functions), clock and interrupt codes, and input/output operation codes. Modular design with extensive documentation has been implemented throughout program development allowing for ease of modification and further extensions to the existing emulation. Emulation and the hardware architecture of the AN/UYK-20 and the Burroughs D-machine, are discussed in conjunction with the AN/UYK-20 emulation itself. Methods of testing and debugging, sample test programs and recommendations for continued design modifications to the emulation are presented.

CONTENTS

I. INTRODUCTION..... 9

A. STATEMENT OF THE PROBLEM..... 9

B. APPLICATIONS OF THE AN/UYK-20 10

C. PROJECT DESIGN OBJECTIVES..... 11

II. EMULATION..... 13

A. HISTORICAL BACKGROUND..... 13

B. MICROPROGRAMMING..... 16

C. THE GOALS OF EMULATION..... 21

D. EMULATION VERSUS SIMULATION..... 21

E. EMULATION TECHNIQUES..... 23

F. EMULATION HARDWARE..... 25

III. AN/UYK-20 ARCHITECTURE..... 28

A. HARDWARE DESIGN..... 29

B. INSTRUCTION FORMATS AND REPERTOIRE..... 38

1. Repertoire of Instructions..... 38

2. Instruction Format..... 41

IV. BURROUGHS D-MACHINE..... 46

A. HARDWARE DESCRIPTION..... 46

1. Logic Unit..... 48

2. The Control Unit..... 52

3. Memory Control Unit..... 53

4. Microprogram memory (M-memory)..... 55

B. NPS MICROPROGRAMMING FACILITY..... 56

1. Physical Description..... 56

2. Input/Output Interface..... 58

3. Memory Interface..... 59

C. MICROINSTRUCTION TIMING..... 60

D. TRANSLANG..... 64

V. AN/UYK-20 EMULATOR..... 66

A. EMULATION DESIGN..... 66

1. Functional Components..... 66

2. Main Memory Organization..... 68

3. Emulation Program Status Word..... 70

4. D-Machine Registers..... 73

B. LOADER..... 75

C. THE FETCH MODULE..... 77

D. OPCODE IMPLEMENTATION..... 83

E. UTILITIES..... 86

F. INPUT/OUTPUT CONTROLLER..... 87

VI. EMULATION TESTING..... 91

A. METHOD OF TESTING..... 91

B. SAMPLE TEST PROGRAMS..... 94

C. TEST RESULTS..... 95

VII. SUMMARY AND RECOMMENDATIONS..... 97

A. EXPERIENCE WITH HARDWARE..... 97

B. LESSONS LEARNED..... 98

C. EMULATION PROBLEMS..... 99

D. RESULTS.....100

E. RECOMMENDATIONS AND FOLLOW-ON TOPICS 101

APPENDIX A. AN/UYK-20 EMULATOR USER'S MANUAL.....103

APPENDIX B. LOADER CONTROL CARD FORMATS.....107

APPENDIX C. AN/UYK-20 EMULATOR INSTRUCTION FORMAT.....110

APPENDIX D. SAMPLE DEBUGGER OUTPUT.....111

APPENDIX E. SAMPLE TEST PROGRAMS.....113

APPENDIX F. EMULATOR LISTING.....120

BIBLIOGRAPHY.....259

INITIAL DISTRIBUTION LIST.....262

ACKNOWLEDGEMENTS

Our sincere gratitude is expressed to Lieutenant Lyle V. Rich, SC, USN for sponsoring this thesis and without whose guidance this research would not have been a success. Technical expertise for the AN/UYK-20 was provided by John H. Westergren, Field Engineer, Mini-Systems Support Operations, Sperry Univac Computer Systems, St. Paul, Minnesota. Initial instruction on the operation and design of the Burroughs D-machine was graciously provided by Lieutenant Jerry M. Haggerty, USN and Lieutenant John M. Hartling, USN. Finally, the authors would like to dedicate this thesis to their wives, whose love, devotion, and understanding enabled this project to be completed.

I. INTRODUCTION

A. STATEMENT OF THE PROBLEM

The Navy has been challenged with maintaining the newest, most efficient tactical data systems consistent with the continually increasing demands and requirements of the real-time environment. There is an extensive conversion effort required to change from existing systems to newer more sophisticated technology such as the AN/UYK-20. Inherent in upgrading to a new system is the complex software redesign and modification process which is often hindered by the absence of the new computer system.

Unfortunately, the demands of a military installation require software generation prior to implementation of an upgraded computer system. One solution to this problem is to utilize for software development an intermediate computer system which has the capability of emulating the anticipated target machine. This provides a vehicle for software design, development, and testing prior to transitioning to the new system.

Currently, the Naval Postgraduate School (NPS) Computer Science Department maintains a Burroughs Interpreter-Based System known as the Burroughs D-machine. The D-machine is capable of being microprogrammed to emulate any of a miriad

of target machines. It effectively enables students to create their own computer knowing only the machine instruction repertoire for the control unit in the target machine.

The problem presented was to develop a feasible working model of the AN/UYK-20 on the microprogrammable Burroughs D-machine. The project provided an opportunity to obtain practical experience with contemporary hardware and to manipulate writeable control stores to imitate a Navy tactical computer.

B. APPLICATIONS OF THE AN/UYK-20

The Univac AN/UYK-20 minicomputer is a general purpose militarized digital computer adaptable to numerous tactical applications. The AN/UYK-20 has been successfully utilized in many time-critical, real-time systems including fire control radar, communication controllers, signal processing analyzers for sonar and beacon signals, and numerous weapons control systems. A subsequent chapter will be devoted to the technical aspects and internal design of the AN/UYK-20.

Because of its size, ruggedness, and computing capabilities, the AN/UYK-20 has been designated the Navy's standard tactical minicomputer [16]. It was selected for emulation in order to provide a feasible platform for software development to those military installations either contemplating or in the process of receiving an AN/UYK-20.

A workable emulation would allow military applications such as data reduction, navigation, telemetry, sensor processing, range tracking and logistics to have software packages developed, tested, and modified prior to arrival of the AN/UYK-20. Furthermore, it would permit personnel to become familiar with the machine by providing advanced training, thereby easing the transition phase to the new system.

C. PROJECT DESIGN OBJECTIVES

Several design techniques were used throughout the development of this project: 1) modularity, 2) structured programming, and 3) extensive documentation. These design features will aid the interested reader as well as simplify any future extensions or modifications to the existing emulation.

Modular design was utilized by creating independent program segments which were individually developed, debugged, and tested. These modules or subroutines provided a strong foundation which were readily modified throughout the entire programming effort. Conceptually, the emulation was divided into relatively small entities which were further reduced to program segments rarely exceeding one page in length.

Structured programming was demonstrated by utilizing a limited number of control flow structures and maintaining a common logical design throughout the entire emulation. Comprehensible code held precedence over extremely efficient

code.

In addition to modularity and structured programming, the entire programming endeavor was supplemented with extensive commenting to provide the necessary self-documentation to promote and facilitate program translation, modification, and fusing with the other independent modules. These concepts promoted the extensive team effort required to achieve the research goals. In addition, it will provide ease of program maintenance and modification in the future.

II. EMULATION

A. HISTORICAL BACKGROUND

The term microprogramming was first utilized in an article by Professor M. V. Wilkes of the Cambridge University Mathematical Laboratory in 1951 [24]. His paper concentrated on a control section within the computer which, when programmatically controlled, performed register-to-register data transfers sequentially and in parallel for the execution of a single machine instruction. A sequence of operations (microinstructions) required for execution of a machine instruction is considered a microprogram.

Traditionally, the computer has been composed of essentially five components: the arithmetic/logic unit, the control unit, memory or storage, input, and output (Figure II-1). The control section sets the proper conditions for the opening and closing of required gates in the logic network. Historically, the control section has been hardware consisting of a series of decoders and flip-flops along with their associated circuitry. Therefore, every machine instruction had a fixed interpretation which was hardwired within the control unit.

In 1957, Wilke's definition of microprogramming was slightly modified. It was defined as a technique of design-

ing the control circuits of an electronic digital computer to interpret and execute a given set of machine operations as an equivalent set of micro-operations [15].

The hardwired control section can be modified by interchanging ROM modules or other hardware components, by replacing the control section with a programmable (dynamically writeable) control store which in itself is a separate word-organized memory (Figure II-2) or by combining both approaches. A programmable control store allows rapid changes in the machine's instruction repertoire while maintaining maximum design flexibility. The resulting computer system is microprogrammable and capable of storing a series of changeable machine personalities.

The computer control store can thus be modified to allow the execution of machine language programs intended for a variety of machine architectures. This process can be compared to replacing hardware components found in conventionally designed computer systems. The primary advantage of microprogrammed logic is the capability to perform various control sequences without hardware modifications. The process through which the hardware components of one machine (host) are made to imitate the specific hardware characteristics of another machine (target) is known as emulation.

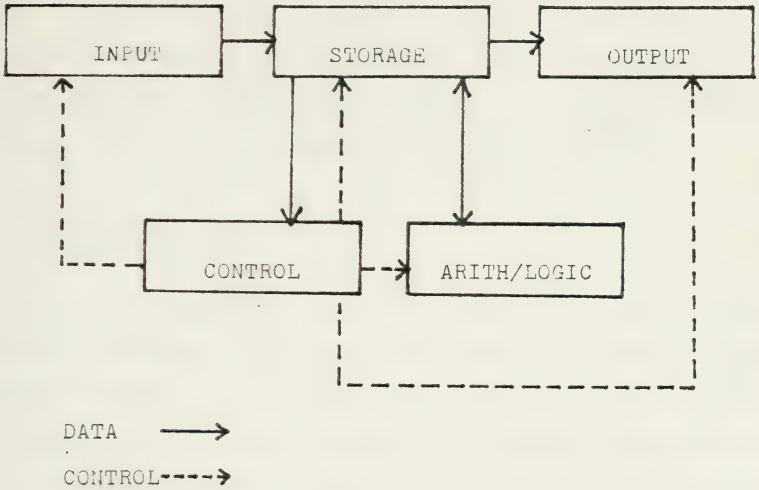


Figure II-1 [11]

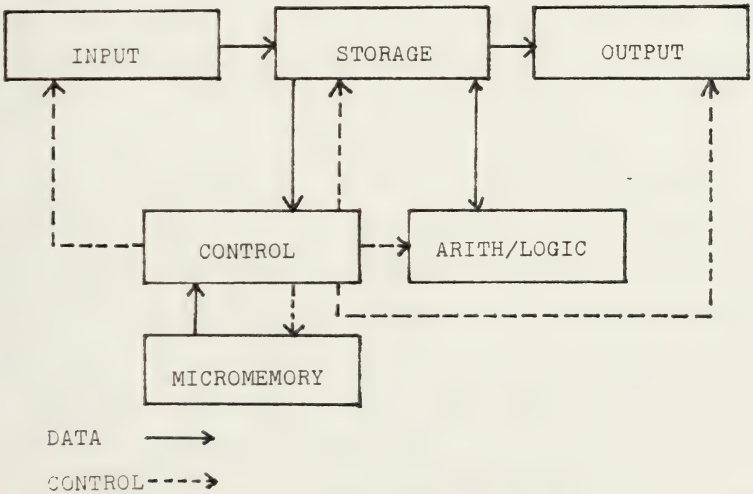


Figure II-2

Emulation allows the computer scientist to create various machine architectures from a single microprogrammable host. The complete set of microprograms (firmware) and the necessary hardware, as well as the required software, added to one computer system enabling it to execute programs designed for another system is known as an emulator.

B. MICROPROGRAMMING

Computer manufacturers have made available numerous microprogrammable machines which permit the user to tailor his instruction repertoire to meet the needs of his particular application. Some examples of microprogrammable computer systems are the Burroughs D-machine, the Nanodata QM-1, the Varian 73, the Standard Logic CASH-8, or the Hewlett-Packard 2100. These microprogrammable systems provide the benefits of flexibility, lower system costs and a systematic approach to system design if utilized effectively.

When a manufacturer designs a dynamically writeable control store, the amount of parallelism to be allowed must be determined. Parallelism is defined to be the simultaneous control of numerous hardware resources. There are basically three forms of control: vertical, horizontal, and residual. In vertical microprogramming, each instruction controls a single operation with program flow being sequential, unless the instruction was a conditional or unconditional branch. By contrast, a horizontally microprogrammed machine is

programmed via instructions which simultaneously control multiple resources including condition testing and microinstruction sequencing.

Horizontal microinstructions usually are not encoded which means each bit controls one machine resource or operation. They usually have a wider word than vertical instructions and consequently consume more memory. Vertical instructions are usually encoded with one or two levels. Encoding means the value of a control field in the microinstruction is a binary code specifying which resource or operation is to be performed. The horizontal microinstructions have the potential of being much more efficient resource managers and consequently are more difficult to optimally design than their vertical counterparts.

Combining the attributes of horizontal and vertical microprogramming results in residual control. This method saves memory by using vertical microinstructions while simultaneously controlling multiple parallel resources via setup registers.

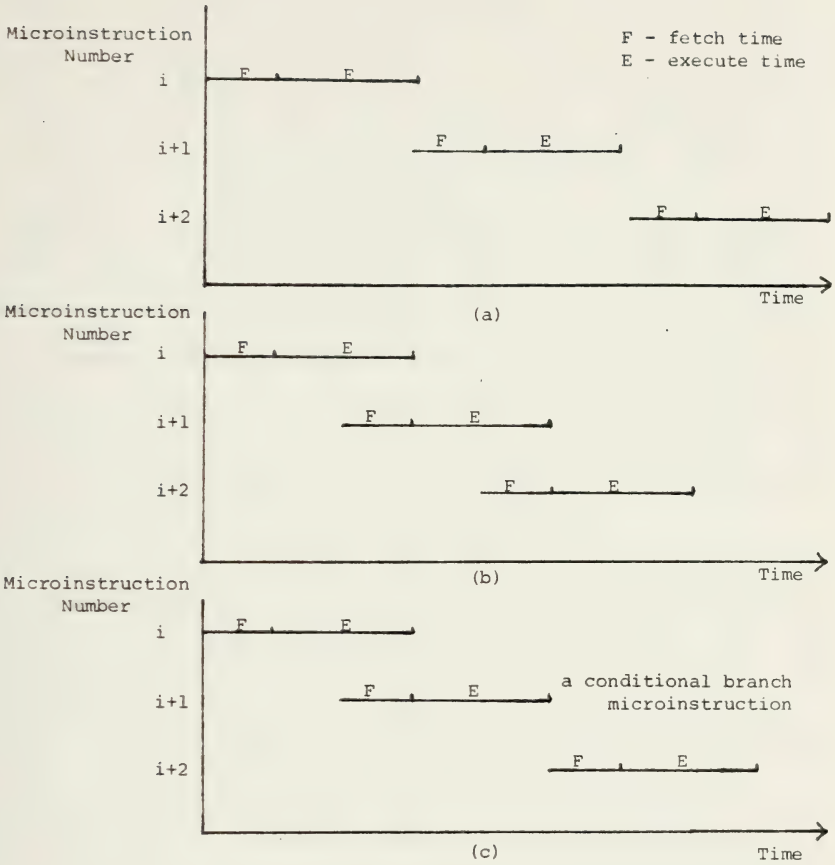
Microinstruction implementation severely effects the speed of microprogram execution. In serial implementation, one microinstruction is fetched and fully executed prior to fetching the next instruction. This technique offers the advantage of logical simplicity while suffering from lack of efficiency since it consumes the maximum amount of time.

'Parallel' implementation permits fetching of the next

instruction before termination of the previous instruction. The obvious advantage is execution speed which is of utmost importance when emulating another machine (Figure II-3) [2].

Another significant microprogramming characteristic is the number of phases used in the execution of each microinstruction. A monophase system means there are no subdivisions of the basic clock pulse and consequently each microinstruction is controlled by the transmission of the leading edge of a clock cycle. In a polyphase implementation scheme, the basic clock cycle is subdivided into minor phases which are independently generated via hardware. Although polyphase operations are more complex and require complicated control, they do permit faster resource manipulation when they are efficiently coded by allowing multiple operations to be performed during the same phase(s).

The microprogrammability of a given computer and the capabilities of its associated microprogramming language are directly effected by the the presence or absence of each of the alternative microprogramming characteristics described above. The microprogramming language spectrum ranges from the lowest level or microlanguage through the assembly languages to the high level procedural languages.



(a) Serial fetch and execute.

(b) Parallel fetch and execute.

(c) Combined serial-parallel where next address depends on conditions in present cycle.

Figure II-3 [2]

The problems of microprogramming can be significantly reduced if suitable software support exists and is readily available. This support is usually in the form of simulators and debuggers. Typically, a simulator provides an alternative to assembly level coding by permitting the user to code in a higher level language and yet achieve the same results at the expense of some added memory and execution time. Debuggers are extremely useful in the developmental stages of microprogramming especially for new and experimental system design. Debuggers permit dynamic access to the machine status and register contents at the instant they are employed, i.e. a trace feature. Some debuggers offer the opportunity of assembling in-line. This option can drastically reduce required debugging time.

The primary application of microprogramming is to implement the necessary control structure required for the analysis and execution of machine level instructions by means of programmed control stores rather than hardwired logic. Therefore, a dynamically microprogrammable computer can provide a software development system which can be a cost-effective approach to experimentation with potential candidates for replacement computer systems or the design of completely new systems to fit the needs of unique applications.

C. THE GOALS OF EMULATION

A well-designed emulation can provide an opportunity to experiment and create software for new computer systems before the actual hardware is available. The utilization of an emulator can almost eliminate reprogramming, consequently smoothing the system transition period. In addition, emulation has provided a workable model of new systems under consideration for procurement, providing a much more detailed cost-benefit analysis of system conversion.

Furthermore, it is often economically sound to emulate a second generation computer with a third generation system. This provides growth to a contemporary system while fulfilling the requirements of the past in a cost-effective manner. However, this can be a disadvantage if the programming staff uses the emulation as a link to the old system and consequently fails to take advantage of the attributes of the new system.

D. EMULATION VERSUS SIMULATION

To accomplish the emulation objectives, certain design features must be incorporated into an emulation. Naturally, execution time and allocated memory are the two foremost considerations. Traditionally, the concept of mimicking another computer has been accomplished by either a simulator or an emulator, two concepts often confused with one another.

A simulator is a series of high level language (HLL) or assembly language statements which individually do not behave like the target machine instructions. The host machine executes its own native instructions in order to imitate the target machine operations. Consequently, simulation is a rather slow technique because it requires an intermediate translation. In addition, simulation of certain instructions such as bit manipulation and shifting operations can require an enormous amount of intermediate code generation demanding a significantly larger memory allocation.

An emulator is a microprogram that is executed on the host machine, performing machine instructions of the target machine. Since an emulator accepts the binary object code of the target machine and directly executes these instructions, it can be extremely efficient in terms of time and space requirements. The execution time of an emulation is dependent upon many factors: clock rates of the two machines, frequency of memory references, high speed shifting compatibility, required register mapping between target and host machines, bit manipulation capability of the two machines, condition code selection and testing, flexible data path selection capability, interrupt similaritites, input/output compatibility, and microprogramming efficiency. If the hardware features between target and host machines are extremely compatible and highly efficient microprogramming has been employed, an emulation performance ratio (host

to target) of nearly one to one can be attained. This emulation performance ratio (EPR) has been demonstrated by the emulation of the SKC-2070 on the Nanodata QM-1 computer. It is possible to achieve an EPR better than one to one under ideal situations, when the host machine has a much faster internal operation execution rate [1].

Several distinct advantages can be realized using emulation as compared to simulation. The execution speed is significantly better by at least an order of magnitude. The target machine representation in firmware is closer to the actual hardware design and total access to the lowest machine level is achievable. Perhaps most noteworthy, emulation provides the opportunity to rapidly create test beds for numerous machine architectures and provide a basis for new system development.

E. EMULATION TECHNIQUES

Traditionally, there have been three approaches for emulating machine instructions: 1) hardware or firmware assistance to a software simulation as demonstrated by the IBM 360/65 emulation of the IBM 7090, 2) independent host system hardware or firmware which provides for complete execution of the target machine's instruction repertoire of which the Burroughs D-machine emulation of the AN/UYK-20 is an example, and 3) an auxiliary processor which is operated in conjunction with the host machine to execute target machine instructions [14].

Software-controlled emulation is usually characterized by categorizing the target machine instructions into three distinct classes: easily emulated instructions, complex instructions not readily emulated, and those instructions not deemed necessary for the desired application. Instruction usage is significant in this classification process. Each class of instructions becomes a candidate for direct hardware or firmware implementation. The first emulated function in this approach is usually the fetch and analysis operation. After the instruction is analyzed, the appropriate opcode subfunction can be executed.

An alternative emulation technique is the firmware-controlled method. This approach is identified by having system control reside completely in firmware or hardware during the emulation process. All instructions are executed on the host machine as if they were indigenous to the target machine. This method is much more efficient than the software-controlled technique; however, it is more expensive and the cost differential is directly related to the required performance level. Performance is dependent upon the number of required data paths, arithmetic units, and other additional logic circuitry which must supplement the host machine architecture.

Upon entering the emulation mode in a firmware-controlled emulator, the machine performs like the target machine until encountering an exit situation. There exists three exit modes: 1) priority interrupt, 2) not implemented

instruction, and 3) deliberate exit because of a debugging routine.

The third emulation technique consists of utilizing auxiliary hardware electronically attached to the host computer for the sole purpose of executing target machine instructions. In effect, a target machine is composed of host machine hardware with the necessary additional components required to create an effective emulator.

F. EMULATION HARDWARE

The development of writeable control stores and microprogramming techniques have significantly influenced computer design. This section will describe some of the available dynamically microprogrammable hardware (Figure II-4).

The Hewlett-Packard 2100 is a general purpose minicomputer. It has a unique control store divided into two segments. One section is ROM and the other section is user programmable. The machine is vertically microprogrammed using a standard 80 instruction machine language repertoire. A debugger and assembler assist the user in microprogram development [2].

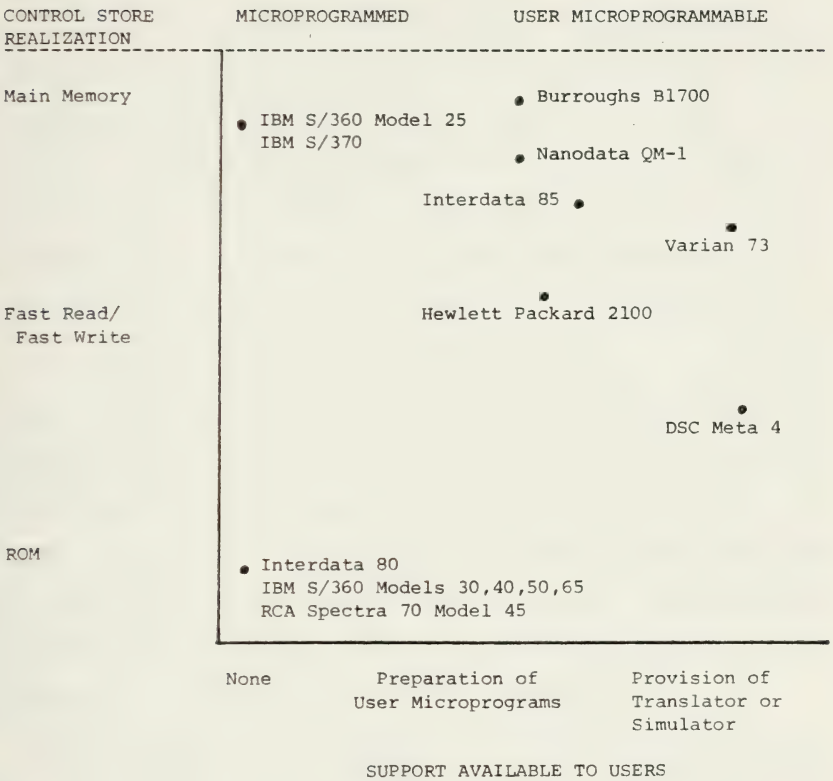
The Standard Logic CASH-8 is a high speed digital controller with a separate control store. It consists of 16 general purpose registers and an accumulator. The CASH-8 is vertically microprogrammed but does not support any language

above microlanguage [2].

The Varian 73 is a general purpose minicomputer that has a 150 instruction set. The horizontal microinstruction consists of 64 bits with 25 fields, some of which indicate register transfers, ALU operations, shifting, control store addressing, condition testing, I/O control and memory operations. The Varian 73 contains both a ROM control store and a writeable control store loadable from main memory. A microprogram assembler and interactive simulator are available [2].

The Nanodata QM-1 is unique in that it contains both a control store and a nanostore which are both loaded under user program control. The 18-bit vertical microinstructions are stored in the control store, fetched and then interpreted under nanoprogram control. A horizontal nanoinstruction is 360 bits which is subdivided into five 72-bit vectors. Assemblers for both microprograms and nanoprograms are available [2].

The previously described machines represent a small sample of the available microprogrammable computer architectures. The availability and flexibility of these computer systems has stimulated demand for these devices. Consequently, hardware manufacturers have been compelled to produce writeable control store equipment to satiate the needs of the computer market.



Note: Relative microprogrammability is the distance from the origin to the machine point in two space.

Figure II-4 [2]

III. AN/UYK-20 ARCHITECTURE

Constructing an efficient emulation requires a precise understanding of the architecture and performance characteristics of the machine being emulated. An emulation must attempt to match the target machine's features and maintain its flexibility of hardware design as closely as possible. Although it is not required, an operational demonstration of the emulated machine can solve many emulation questions.

In emulating the AN/UYK-20, architecture and performance criteria were derived from technical publications, since an actual machine was unavailable. When inconsistencies appeared in the documentation, specific questions were posed to a UNIVAC field engineer, who often tested programs on the AN/UYK-20 to resolve inconsistencies. Documentation coupled with an expert consultant provided sufficient information for emulating the AN/UYK-20 successfully.

The intent of this chapter is to outline those features of the AN/UYK-20 significant to the emulation. A detailed hardware description can be found in Refs. 20, 22, 28.

A. HARDWARE DESIGN

The AN/UYK-20 was designed for the Navy to fulfill the requirements for small or medium size general purpose data processing in shipboard, mobile shelter, or other military environments. Sperry Univac incorporated minicomputer technology in constructing the AN/UYK-20, including MSI circuitry design, microprogrammed control, memory modularity, and asynchronous or synchronous input/output channels.

The AN/UYK-20 had to be extremely flexible in its applications, offering a wide range of configuration possibilities which were derivatives of the basic design. Modularity, a concept highly desirable in a military environment, was achieved by offering options that could be easily added using printed circuit cards and/or memory modules.

The AN/UYK-20 can accomodate up to eight 8K, sixteen-bit word boards of magnetic core storage with an access time of 750 nanoseconds. The central processor is controlled by a programmable micromemory which can be expanded by an additional 512 words. The microprogram controller is programmed at the factory, but the additional micromemory option is user defined. Both sections of micromemory are programmed using fusible links, and once programmed they are completely static (Figure III-1).

A memory interface is responsible for the transfer of data to memory from the central processor (CP) and input/output controller (IOC). Both the IOC and the CP are

capable of accessing all of memory (65,536 words maximum). The addition of direct memory access (DMA) provides a second memory interface and an additional access port which is connected to each of the two 32K memory segments.

The input/output controller permits the central processor to communicate with the external devices without interfering with program execution. The IOC has a maximum of 16 parallel or serial channels. Parallel data transfer takes place asynchronously using 8-bit, 16-bit, or 32-bit transfers. Serial interfaces are either synchronous or asynchronous, with word-to-serial or serial-to-word conversions occurring in the IOC. The IOC and CP compete for memory access through the memory interface with priority given to the IOC in the event of a simultaneous request. The IOC is permanently assigned several memory addresses for command word and interrupt word storage.

The addressable high-speed registers available in the AN/UYK-20 include the program address register (P-register), two 16-bit status registers (SR1 and SR2), a real-time clock register (32-bits), a monitor clock register (16-bits), and a set of sixteen 16-bit general registers. An additional stack of 16 general registers is an available hardware option.

The sixteen general registers were included to enhance the speed and performance of the AN/UYK-20, allowing most programs to use a great proportion of register-to-register

instructions. These general registers can be used as accumulators for arithmetic, shift, or logical functions, as index registers, or as temporary storage locations. The second set of general registers can be readily employed via a status bit. This status bit designates which general register stack is to be utilized. The duplicate set of general registers yields dividends in a multi-task or heavy-interrupt processing environment. This additional register set can be used to provide high-speed temporary storage, thus avoiding slower main memory storage of working variables.

The two 16-bit status registers and the program address register represent the machine status of the AN/UYK-20. When these registers are collectively referenced, they are called the program status word (48-bit PSW). The P-register indicates the next instruction to be executed. This instruction may be a 16-bit single-word instruction or a 32-bit double-word instruction. Program control can be modified by using an instruction which manipulates the contents of the P-register.

Status register 1 contains bit information concerning condition code settings, overflow, and carry bits, interrupt codes, and numerous other machine indications (Figure III-2).

AN/UYK-20 FUNCTIONAL ARCHITECTURE

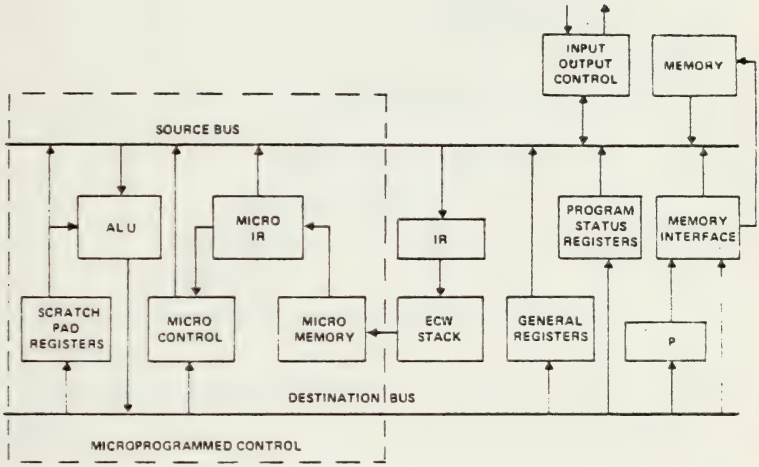
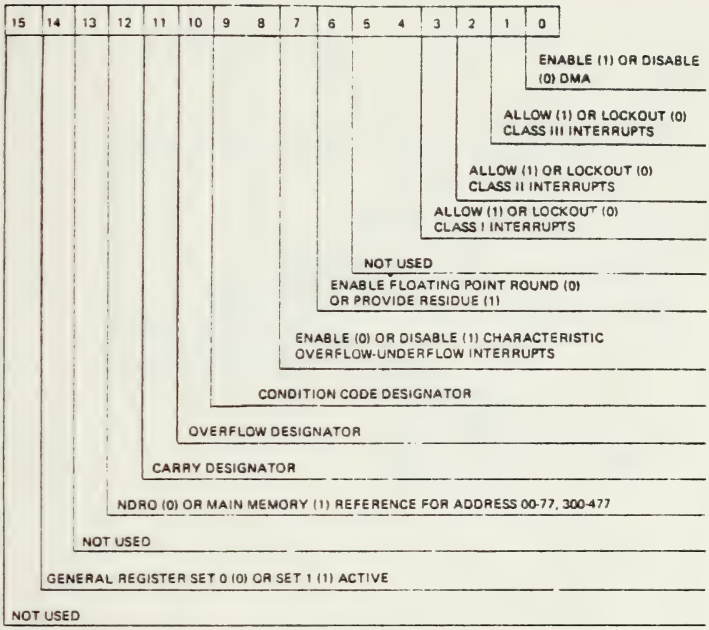


Figure III-1 [22]

STATUS REGISTER 1



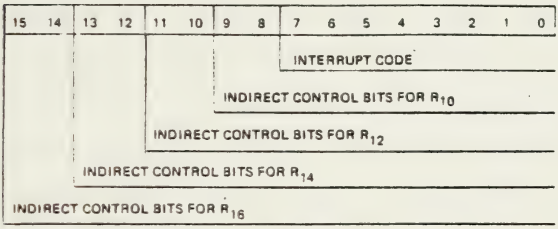
CONDITION CODES

SR BITS 8 and 9	ARITHMETIC	COMPARE
00	0	$(R_a) = (R_m)$ or (Y)
01	>0 (POS)	$(R_a) > (R_m)$ or (Y)
10	Not Used	Not Used
11	<0 (NEG)	$(R_a) < (R_m)$ or (Y)

Figure III-2 [23]

Status register 2 holds control bits for direct or indirect addressing, and holds interrupt codes. Interrupt processing routines set bits in the interrupt code field corresponding to the IOC interrupt (Figure III-3).

STATUS REGISTER 2



INDIRECT CONTROL BITS	MEANING
00	Normal Addressing.
01	Normal Addressing.
10	Indirect Addressing w/o indexing; IW1 at Y = y.
11	Indirect Addressing with indexing; IW1 at Y = y + (Rm).

Figure III-3 [23]

The real-time clock and monitor clock registers provide program-controlled interrupt capability which is useful for timing and synchronizing program segments with real-time events. The real-time clock (RTC) is a 32-bit register used as count-up storage while the monitor clock (MON) is a 16-bit count-down register. A one kHz internal oscillator controls the counting speed of both registers. An optional

external clock operating at a frequency up to 50 kHz is also available.

Interrupt processing in the AN/UYK-20 is conducted using a priority level scheme which classifies interrupts into three priority levels (classes). Interrupts within the same class are assigned a priority ranking and a code which identifies which processing routine to execute. During interrupt handling, all interrupts of the same level or lower level are locked out until the CP is completed processing the current interrupt. Higher priority interrupts can override the lockout and cause the CP to honor them first, holding the lower level interrupts in abeyance until higher level interrupt processing is completed. The highest priority interrupts are hardware malfunctions, followed by software interrupts, and at the lowest level, IOC interrupts.

Permanent locations in memory corresponding to each interrupt class hold the PSW and RTC when an interrupt is being honored. Likewise, other permanent memory addresses assigned to each interrupt class hold the appropriate interrupt routine entrance location to be loaded into the PSW.

Memory addressing is accomplished using 64 page address registers which separate memory into 1024-word pages. Absolute addresses are formed by isolating the upper six bits of the relative address to find the page address register number, and then concatenating the lower six bits of the

page address register contents with the lower ten bits of the relative address (Figure III-4). Any operation that stores into memory sets the most significant bit of the page address register used in generating the address.

Some additional hardware features of the AN/UYK-20 are those functions available on the maintenance panel of the machine itself. These include a breakpoint feature which allows an operator to insert from the panel an address which causes the AN/UYK-20 to stop execution when the selected address is referenced. Other available toggles allow halting execution programmatically using Key 1 or Key 2 on the maintenance panel. These additional hardware features are useful debugging tools.

MEMORY ADDRESS GENERATION

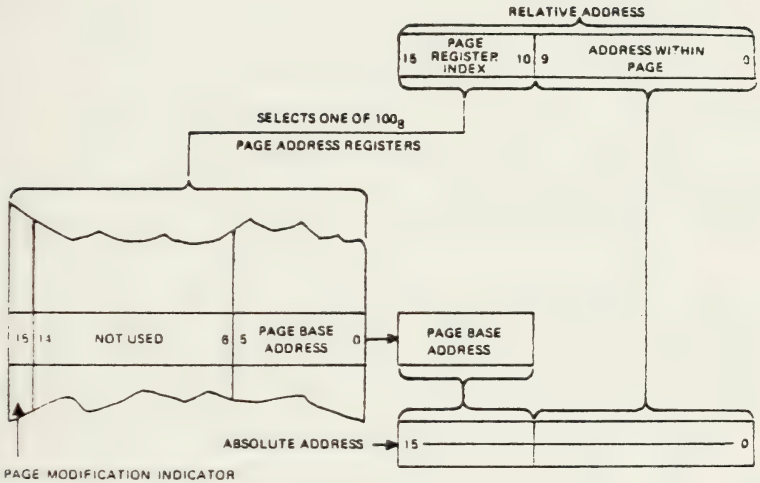


Figure III-4 [22]

B. INSTRUCTION FORMATS AND REPERTOIRE

1. Repertoire of Instructions

The AN/UYK-20 instruction set is composed of nearly 260 separate instructions designed to be both versatile and comprehensive. Both single-word (16-bit) and double-word (32-bit) instructions are available. Some of these instructions are specifically designed to meet the requirements of a real-time environment. A few sample instructions include:

a. Local jump - used to facilitate loops, saving several steps in program execution.

b. Reverse register - used in a communication environment when data is received in one sequence but must be transmitted in reverse sequence.

c. Set bit, clear bit, and test bit - used to test individual bits in registers saving considerable execution time in programs that communicate via flags and status words.

Additional flexibility is provided when the 'math pac' hardware option is included in the AN/UYK-20 configuration. Some 33 additional opcodes are added to the instruction set in order to increase the computational capabilities of the machine. An instruction for square root, double precision multiply/divide instructions, as well as hardware trigonometric and hyperbolic functions which utilize coordinate conversion algorithms (cordic) are included.

Single-word instructions are generally employed when manipulating operands in high-speed registers. Double-word instructions are used in operations involving direct or indirect addressing with or without indexing, or supplying 16-bit constants to programs. Typical instruction speeds for a single-word versus a double-word instruction are:

	SINGLE-WORD	DOUBLE-WORD
ADD	.75 - 1.5 usec	1.5 - 2.25 usec
LOAD	.75 - 1.5 usec	1.5 - 2.25 usec
MULTIPLY	3.8 - 4.0 usec	4.4 - 4.6 usec
DIVIDE	6.8 - 7.0 usec	7.4 - 7.5 usec

Nearly all instructions affect condition bits in status register 1. The AN/UYK-20 sets these bits as a result of two types of operations. Most instructions that do not involve compare logic are categorized as arithmetic instructions. When the result of the arithmetic operation is determined and is about to be stored in memory or a register, a condition code is set indicating whether the result is positive, negative, or zero. Compare instructions set the condition bits based on a greater than, less than, or equal to comparison of two registers or a register and the contents of a memory address.

Two other bits in SR1 are set as a result of computational or shifting instructions. The overflow designator is set whenever an arithmetic or shift operation requires more bits than are provided for in a register (16 bits).

The carry designator is set when an arithmetic operator generates a carry beyond the most significant bit of the register.

The AN/UYK-20 allows five different types of operand formats: single-length, byte, literal, optional floating point, and double-length. Single-length operands are 16-bit values with the sign bit assumed to be in the most significant bit. In arithmetic calculations, the single-length word is assumed to be a two's complement integer. Byte operands are considered 8-bit unsigned integers, and can be the most significant or least significant half-word of a memory location. Literal operands are 4-bit unsigned integers denoted by the 'm' field of a literal type instruction. Floating point operands are formed using two adjacent registers or memory locations with fields containing the sign of the fraction, the characteristic, and 24 bits of the fraction.

Double-length operands are concatenated from two adjacent registers or two adjacent memory addresses. The most significant half of the double-length operand is contained in the low-order register or memory address, and the least significant half in the next sequential register or address. The sign bit for both words resides in the high-order bit position of the most significant half-word and when used in an arithmetic calculation, the double-length operand is treated as a two's complement integer. Instructions involving double-length operands require the low-order

register or memory address to be even, since the adjacent cell address is formed by 'OR'-ing a 1 in the least significant bit of the address.

2. Instruction Format

The AN/UYK-20 has five different instruction word formats, designated by two-letter mnemonic codes. These codes are: RR (Register-Register), RL (Register-Literal), RI (Register-Immediate), RK (Register-Constant), and RX (Register-Index). Each of these formats are designated in the instruction word by a combination of the opcode field and the subfunction code. Registers are identified in the 'a' and 'm' fields of the instruction, and are referred to by the notation Ra and Rm. The 'y' field is treated as an address or arithmetic constant, depending on the instruction (Figure III-5).

The format RR single-word instructions perform operations involving the registers specified by the 'a' and 'm' fields. No memory references are made to access operands, causing considerable savings in execution time. The 'a' or 'm' field may be used to index special operations on registers.

Format RL instructions use a 16-bit instruction word, using one or two general registers. The 'a' designator selects the general register Ra or register pair Ra, Ra + 1. The 'm' designator contains the 4-bit literal value which will be used in the instruction.

INSTRUCTION FORMATS

RR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Op*						0		a				m			

a selects R_a; m selects R_m.

RL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Op*						0		a				m			

a selects R_a; m contains 4-bit constants.

RI-1
Local
Jump

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Op*						1		d							

d is signed number of instructions to jump, relative to current position.
(+ is forward; — is backward)

RI-2
Indexed
Memory

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Op*						1		a				m			

a selects R_a; m selects R_m. R_m selects memory address.

RK

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Op*						2		a				m			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
y															

a selects R_a; $m \neq 0$ selects R_m, m = 0 selects 0.
Operand = $y + R_m$ or 0.

RX

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Op*						3		a				m			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
y															

a selects R_a, $m \neq 0$ selects R_m, m = 0 selects 0.
$y + R_m$ or 0 selects memory address.

*Op is operation code

Figure III-5 [23]

RI format single-word instructions are separated into two catagories. RI Type 1 instructions are local jump instructions where the P-register is increased or decreased by the 'd' field in the instruction. The 'd' field represents the two's complement deviation value and allows the P-register to be altered a maximum of +177 octal or -200 octal. RI Type 2 instructions involve operations that use the general registers Ra and Rm, and a memory address specified by the contents of Rm.

The format RK instructions contain 32 bits of information. The upper 16-bits contain the operation code and the designator fields. The 'a' field selects the general register Ra, and the 'm' field denotes how the next word, containing 'y', is to be used. If 'm' equals zero, then the effective operand address Y, equals 'y'. If 'm' does not equal zero, then Y is formed by adding the contents of Rm to 'y'.

Format RX are double-word instructions similar to the RK instruction that use direct and indirect addressing techniques determined by the 'm' field. If the 'm' field equals zero, then direct addressing without indexing is employed, with the effective operand address formed from the 'y' field itself. If the 'm' field equals 1 through 7, 11, 13, 15, or 17 (octal), then direct addressing with indexing is employed. The effective operand address is formed by adding the contents of Rm to 'y'. An 'm' field value of 10, 12, 14, or 16 (octal) indicates indirection is to be

employed, and the indirect address control fields in status register 2 contain information which is used to generate the effective operand address or a pair of indirect words. When the control fields equal 0 or 1, then direct addressing with indexing results. If the control field equals 2, then the contents of the first indirect word (IW1) is located at 'y'. Finally, if the control field equals 3, then IW1 is located at 'y' indexed by the contents of Rm. Indirect word format is shown in Figure III-6. Cascaded indirection is possible provided that the indirect words are properly encoded.

Byte addressing is accomplished using RX format instructions. A byte identifier (B) is used to determine which half-word (8-bit) is to be referenced. If B equals 0, then the most significant half-word in address Y is the operand byte. If B equals 1, then the least significant byte at Y is the operand. The least significant bit in the indexing register is used as the byte identifier, and the remaining 15 bits are used as the indexing value for finding Y. Indirect byte operand addressing formulae are included in Figure III-6. The following formulae apply for direct byte operand addressing:

m=0, Y=y, and B=0

m=1-7, 11, 13, 15, or 17 octal

Y=y + (Rm)/2 and B=LSB of (Rm)

The preceding section has described the fundamentals of the AN/UYK-20 instruction formats. References 21 and 23 should be consulted for further information concerning instruction formats and decoding.

INDIRECT WORD FORMATS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
NOT USED	I	J		NOT USED								X				IW 1
IW 2																IW 2

OCTAL J VALUE	ADDRESS DETERMINATION
0	Y = IW2; if byte mode, upper byte is used.
1	Y = IW2 + (Rx); if byte mode, LSB of Rx determines byte. *
2	Y = IW2 + (Rm); if byte mode, LSB of Rm determines byte. *
3	Y = IW2 + (Rm+1); if byte mode, LSB of Rm+1 determines byte. *

I = 0, DIRECT ADDRESSING, OPERAND AT ADDRESS Y

I = 1, CASCADED INDIRECT ADDRESSING, NEW INDIRECT WORD 1 AT ADDRESS Y

* To determine the operand address when in byte mode, the contents of R_x, or R_m, or R_{m+1} are right shifted 1 bit position and zero filled in the left most position

Figure III-6 [23]

IV. BURROUGHS D-MACHINE

A. HARDWARE DESCRIPTION

The microprogramming facility at the Naval Postgraduate School is composed of a Burroughs Interpreter-Based system. This system possesses the characteristic of being dynamically microprogrammable and is designed using a simple building block structure. A typical system is made up of a number of interpreters (processors), main memory modules, and input/output devices, along with a switch interlock device (SWI) controlling data flow on the data bus connecting the interpreters to main memory and peripheral devices. The heart of the system is the interpreter, also referred to as the D-machine.

A D-machine possesses five functional modules: the logic unit (LU), the control unit (CU), the memory unit (MU), nanomemory (NM), and microprogram memory (MPM). The system presently installed in the Computer Science Department combines nanomemory and microprogram memory into one functional unit. The architecture of the D-machine is designed around 8-bit word slices. Word lengths from 8 bits to 64 bits are permissible using the same functional unit (Figure IV-1).

INTERPRETER BLOCK DIAGRAM

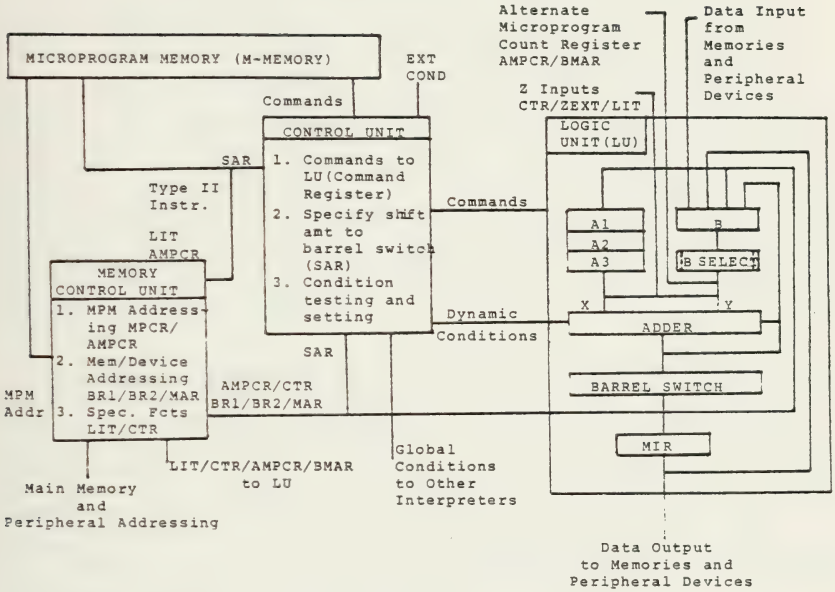


Figure IV-1 [6]

The D-machine used for this thesis has been configured as a 32-bit word processor. Reference 17 provides a thorough and concise description of the architecture of an interpreter-based microprogramming system. Reference 6 details the specifics of D-machine microprogramming which must be thoroughly understood by the programmer.

1. Logic Unit

The D-machine's logic unit performs shifting, arithmetic, and logic functions and contains scratch pad registers and data interfaces for the switch interlock. All adder operations are performed using two's complement arithmetic, and shifting is accomplished in a matrix of gates called the barrel switch.

The scratch pad registers A1, A2, and A3 are identical in function. They act as temporary storage registers within the D-machine and serve as primary inputs to the adder. The control unit determines which register will be an input to the adder. Any of the A-registers can receive the output from the barrel switch.

The B register functions as the primary external input interface from the switch interlock. It acts as the second input to the adder and can receive certain direct inputs from the adder's arithmetic operations. The B register can be loaded from any of the following sources:

a. The barrel switch output.

b. The adder output.

c. External data from the switch interlock (or control panel switches).

d. The memory information register (MIR).

e. The complements of four bit or eight bit carries.

f. The barrel switch ORed with the adder, external data from the switch interlock, or MIR.

The output of the B register is filtered prior to gating into the adder. The 'filter' consists of true/complement selection gates which are divided into three sections: the most significant bit, the least significant bit, and all the remaining central bits. The output of this filter is the independent result of these sections and may be either all zeroes, all ones, or the true contents or one's complement of the contents of the respective bits of the B register.

The memory information register is the interface to the switch interlock. MIR functions as a buffer to main memory or to a peripheral device. It is an output destination of the barrel switch and its output can be sent to the B-register or the switch interlock.

The adder in the logic unit performs all arithmetic functions and can be categorized as a version of a carry look-ahead adder. The control unit can gate various combi-

nations of A, B, and Z inputs to the adder. An 'A' input is defined as an input from one of the three scratch pad registers. A 'B' input is the output of the B register's true/complement filter gates. A 'Z' input is an external input to the logic unit and can originate from one of the following sources:

a. The output of the counter in the memory control unit (MCU) which is gated to the most significant eight bits of the adder with the remaining bits zeroed.

b. The output of the literal register in the MCU which is gated to the least significant eight bits of the adder with the remaining bits zeroed.

c. An optional input which is gated into the middle bytes of the adder with the most and least significant bytes zeroed.

d. The output of the alternate microprogram count register (AMPCR) in the MCU which is gated into the least significant 12 bits of the adder (13 bits for 8K micromemory machines) with all other bits being zeroed.

e. All zeroes.

Two inputs, selected from the A, B, or Z sources, are always gated to the adder. These inputs are referred to as X-select and Y-select. An X-select input may be either an A input or a Z input. If it is not specified, it is

assumed to be zero. A valid Y-select has either a B, Z or 1 as its input. Some Z inputs, however, are valid only as Y-select inputs. Any combination of valid X-selects and Y-selects are permissible addends, with an option of adding a one to the least significant bit. For subtraction operations, the value to be subtracted must be a Y-select; the Y-select input is subsequently two's complemented and gated to the adder. All binary Boolean operations between two adder inputs are allowed, and dynamic condition bits for overflow (AOV), all bits true (ABT), and most/least significant bit test (MST/LST) are available to the control unit for testing. Bit testing is a valuable feature for decoding instruction words.

The barrel switch is a matrix of gates that accepts parallel data from the adder and shifts the data any number of places to the left or right with zero fill. It also can right shift the word in an end-around fashion. The output of the barrel switch may be directed to any of the following destinations simultaneously:

a. The A registers.

b. The B register.

c. The memory information register.

d. The least significant 16 bits to the MCU registers.

e. The least significant three to six bits to the control unit shift amount register (bit length depending on the word size of machine).

2. The Control Unit

The control unit has five major sections: the shift amount register (SAR), the condition register (COND), part of the control register (CR), the decoder for microprogram memory content, and clock control. This module of the D-machine manages the functions of the processor, and is directly responsible for logic unit operation.

SAR and its associated logic control shifting operations by loading shift amounts into SAR and generating the required controls indicated by the current microinstruction for the barrel switch. In addition, the SAR's logic generates the 'word length complement' of the SAR contents where the complement is defined to be that amount which would restore the bits of a word to their original position after an end-around shift of N followed by an end-around shift of the 'complement' of N. For example, if an end-around right shift of 20 was required in a 32-bit D-machine, another end-around shift of the complement of 20 (12) would be required to restore the contents to its original value.

The condition register has four major functions:

a. It stores 12 resettable bits which are used as error indicators, interrupts, status, and lockout indicators.

b. It selects one of 16 condition bits for performing conditional operations. These 16 bits are composed of the 12 condition bits of the

condition register plus the 4 dynamic conditions generated by the LU adder during the present clock time.

c. It decodes bits from the memory for resetting, setting, or requesting the setting of designated bits of the condition register.

d. It resolves priority between interpreters in the setting of global condition bits (GC), thereby providing a method of controlling inter-interpreter lockout.

The control register stores the control bits of the 56-bit microinstruction that are not being used in the first phase of the execution cycle. The control register is subdivided into sections which are used by the memory control unit, the logic unit, and the control unit during the execution phase of a microinstruction. For a description of timing and phases, see section C of this chapter.

3. Memory Control Unit

The memory control unit provides the basic addressing interface between the D-machine and both main memory (S-memory) and microprogram memory (M-memory). One MCU can address 64K words (256K bytes) of main memory, and if the D-machine is configured with a second MCU, a maximum of 128K words can be addressed.

The memory control unit has three major sections:

a. The microprogram address section controls the addressing of microprogram memory and the sequencing of microinstructions. It contains the microprogram count register (MPCR), the alternate microprogram count register (AMPCR), the incrementer, the microprogram address controls register, and their associated logic. For standard 4K M-memories, the MPCR and AMPCR are 12 bits long. For 8K M-memories, MPCR and AMPCR are 13 bits in length (the D-machines installed at the Postgraduate School employ 8K M-memories). AMPCR is a Y-select input to the logic unit adder.

b. The memory/device address section contains the 8-bit memory address register (MAR), two 16-bit base registers (BR1 and BR2), output selection gates, and associated control logic. When forming a memory address, the lower eight bits of a base register and MAR are concatenated. The concatenated 24-bit contents of BR1/ BR2 and MAR (BMAR) is a valid Y-select input to the logic unit adder.

c. The Z register section contains two registers which are Z inputs to the logic unit adder. The literal register (LIT) is an 8-bit register into which constants are loaded. An 8-bit counter (CTR) is used in conjunction with a counter overflow condition bit to control iterative

looping. The Z register section also contains selection gates for the loadable counter and its associated logic.

4. Microprogram memory (M-memory)

A D-machine may have a dual or single microprogram memory scheme. As indicated earlier, the D-machines used in this emulation project had a single microprogram memory, consolidating the microprogram memory and nanomemory into one 56-bit programmable store memory, often referred to as M-memory. Microprograms, consisting of 56-bit microinstructions, are dynamically changeable by the user, thus distinguishing the D-machine as an extremely flexible computing device.

The sequencing of microprogram instructions is controlled by a condition bit procedure which determines the successor command to be executed. M-memory provides data to the condition testing logic which then determines which condition is to be tested. The output of the condition testing logic is a true/false signal that is gated to the successor selection logic. This logic then selects between the three true and three false successor bits also provided by the M-memory word. The three selected bits provide eight possible successor combinations:

a. WAIT Repeat the current instruction.

b. STEP Step to the next instruction.

c. SKIP Skip the next instruction.

d. JUMP Jump to another M-memory address.

e. RETN Return from a microprogram subroutine.

f. CALL Call a microprogram subroutine.

g. SAVE Step and save the instruction address.

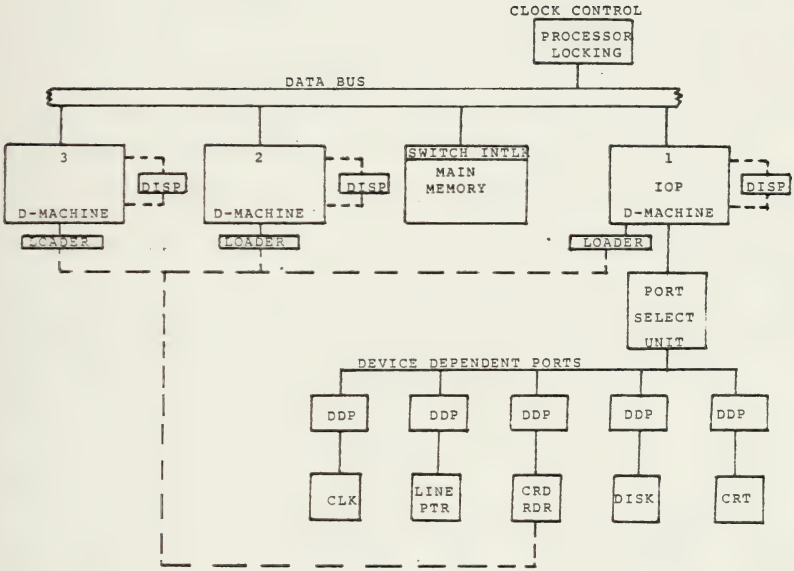
h. EXEC Execute one instruction out of sequence.

The particular successor command chosen controls gates which select the appropriate M-address from MPCR or AMPCR and provides incrementing logic for generating the next M-memory address. Except for the EXEC command, the MPCR is loaded with this M-memory address.

B. NPS MICROPROGRAMMING FACILITY

1. Physical Description

The Computer Science Department of the Naval Postgraduate School possesses a Burroughs Interpreter-Based System consisting of three interpreters (processors) also known as D-machines, a 64K 32-bit word, main memory module, a card reader, a dual cartridge disk drive, a line printer, and a Datamedia 2500 CRT functioning as a supervisor's console (Figure IV-2). All input and output is performed through a single D-machine processor, hardware configured with device dependent ports (DDP) for peripherals and the external clock.



NAVAL POSTGRADUATE SCHOOL MICROPROGRAMMING FACILITY

BURROUGHS INTERPRETER-BASED SYSTEM

Figure IV-2

After initial light-off and bootstrap, the system is configured into two Burroughs 6700 - LIFO ALGOL Stack-machines, each addressing 32K of memory and each communicating with the input output processor (IOP) in a pseudo-multiprocessing environment. Software, written in ALGOL, is provided which runs on the B-6700 system. A resident monitor control program and disk file manager control the maintenance of system files and the execution of jobs in a batch environment. Both D-machines compete for system jobs input from the card reader or CRT. Other software available includes an ALGOL compiler (a derivative of ALGOL 60), a microprogram translator called TRANSLANG, a line editor, and a simulation program for microprograms. TRANSLANG provides the medium by which microprograms are written. User microprograms loaded into a D-machine change its identity and destroy the Stack-machine previously loaded.

2. Input/Output Interface

Pivotal to the operation of the Burroughs system is the input/output interface. Only one processor, the IOP, communicates with peripherals, and the other D-machines, configured as Stack-machines, must compete for its services. The IOP communicates asynchronously, using a conventional 'handshake' method. Since all interpreters have access to the main memory module, a communications link has been established using the upper two 32-bit words of main memory. If a Stack-machine wishes to communicate with the IOP, it places a message into address 65,535 known as the 'mailbox',

and issues an interrupt (INT) to the IOP. The IOP periodically tests INT, and if set, will retrieve the contents of the mailbox (and mailbox - 1 if required) and perform the desired operation. The other Stack-machine is locked out from interrogating the IOP until it has completed processing the request. Normally, the operation requires transferring a buffer to some output device. When the IOP has completed honoring the request, it places a completion code in the mailbox and sets INT for the Stack-machine requesting the I/O. This interpreter must halt execution and check mailbox to see if the I/O was performed successfully, completing the handshaking process. This protocol permits both Stack-machines to perform input/output independently of each other, provided both maintain strict memory boundaries.

Another function of the IOP is interfacing character code formats between the peripherals and the other two D-machines. When the machines are configured as Stack-machines, characters are passed to the IOP in 6-bit BCL (Burroughs Common Language). The IOP must convert this character set to ASCII for output to the line printer and CRT. A similar translation must be made for input data converting from either EBCDIC or ASCII depending on whether the input source is the card reader or the CRT.

3. Memory Interface

Since all three D-machines must share the 64K of main memory, a priority scheme was developed to resolve

memory reference conflicts. The main memory module is actually a single-ported, 32-bit word, core memory which can be made to appear multiported using a switch interlock unit (SWI) developed by Burroughs. The switch interlock controls the main data bus of the system, and resolves conflicts using a priority scheme. The D-machine with the highest priority is the IOP, with the priority of the other two machines being relative to their physical proximity to the IOP. Once a memory reference has been made, a D-machine may continue execution without waiting for a completion signal from the switch interlock. Although this technique of memory referencing minimizes unnecessary delay, it restricts the program from changing the read or write addresses or the content of MIR (write only) prior to a completion signal.

C. MICROINSTRUCTION TIMING

The Burroughs D-machine initiates a microinstruction once every clock cycle. The D-machines utilized for the AN/UYK-20 emulation operated from a one mHz internal clock, which produced a clock pulse once every microsecond. A D-machine designed with an eight mHz clock, emitter-coupled logic (ECL), and a faster memory cycle time, however, could execute eight times faster. This implies that advances in circuit technology can permit emulations to achieve improved speed and performance with no change in the microprograms.

Every microinstruction is executed using one or more sequential time periods, called phase 1, phase 2, and phase

3. A phase is a constant interval of time equivalent to one clock duration measured from the trailing edge of each successive clock pulse. Some microinstructions only require phase 1 to complete execution. Some require phase 1 and phase 3, and still others require phases 1,2, and 3. A new microinstruction is initiated at each clock cycle, allowing for overlapping of microinstruction execution in phase 1 and phase 3.

Microinstructions consist of two types. In a type 1 microinstruction, events can take place in all three phases:

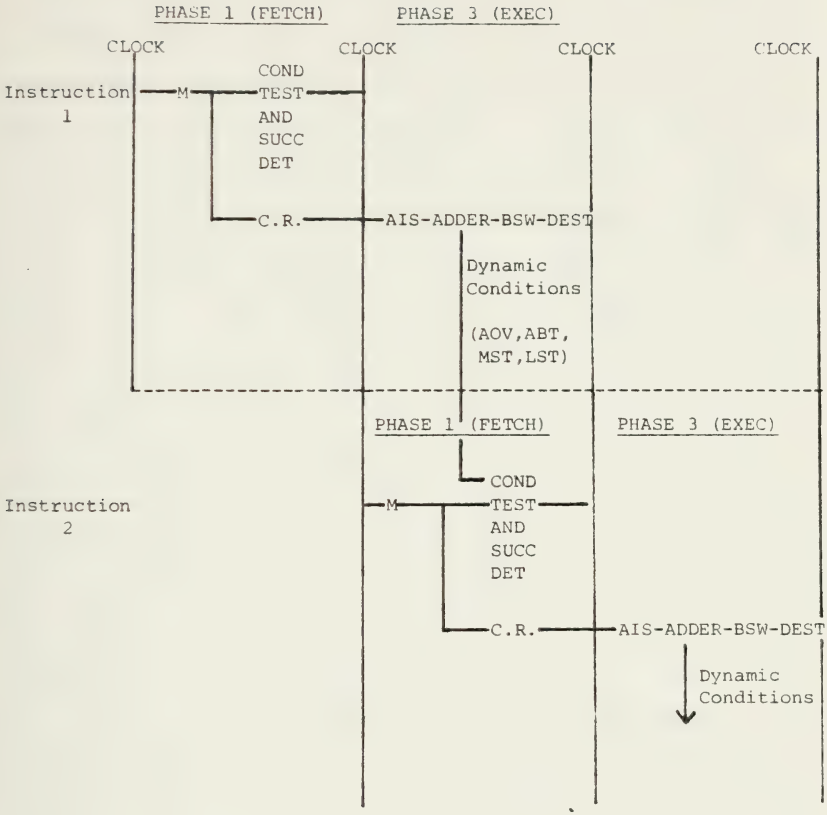
Phase 1: condition testing, (conditional) external operations or (conditional) logic operation initiation after completion of a prior logic operation, and successor M-memory address control.

Phase 2: a holding phase for phase 3 logic operation controls.

Phase 3: the completion phase for logic unit operations and destination register gating specified by logic operation.

During the optional phase 2 period, a type 1 microinstruction execution completion is held in abeyance while a subsequent type 2 instruction is executed. A type 2 microinstruction requires only phase 1 to complete execution, and involves literal assignments to three registers: LIT, SAR, and AMPCR. Phase 2 may also be initiated if the

next sequential type 1 instruction does not execute its conditional logic operation and therefore can complete its execution in phase 1 (Figure IV-3). Appendix D of Ref. 6 has a complete discussion of microinstruction timing.



M - MPM ACCESS TIME
COND TEST AND SUCC DET - CONDITION TEST AND SUCCESSOR DETERMINATION
BSW - BARREL SWITCH
DEST - BARREL SWITCH OUTPUT DESTINATIONS; I.E., REGISTERS (B,CTR,ETC.) AND THEIR INPUT LOGIC
C.R. - COMMAND REGISTER AND ASSOCIATED LOGIC
AIS - ADDER INPUT SELECTION FROM COMMAND REGISTER

Timing Analysis, Type I Instructions

Figure IV-3 [17]

D. TRANSLANG

Microprogramming on the D-machine is accomplished using a microtranslator/assembler called TRANSLANG. TRANSLANG allows the programmer to write microinstructions mnemonically without concentrating on the bit patterns that compose the microinstructions themselves. TRANSLANG is written in ALGOL, the language of the B6700 Stack-machine. Nearly the entire language of TRANSLANG is composed of reserved words recognized by the ALGOL program. Each reserved word has a special meaning which causes the translator to construct particular microinstructions. A TRANSLANG instruction is equivalent to one microinstruction consisting of the set of parallel D-machine functions performed during the clock phases. TRANSLANG is a free form language and instructions may be written in almost any order. Multiple instructions may appear on a line, separated by a period '.'. TRANSLANG constructs include iterative mechanisms, input/output, assignment functions, control transfers, and Boolean, and computational operations. In addition, TRANSLANG permits label definitions and symbolic references for program control flow. Reference 6 is the programming manual for TRANSLANG and contains the complete syntax for the language. Appendix A of Ref. 10 documents additions to the TRANSLANG instruction repertiore.

A microprogrammer may construct complicated microinstructions that perform many different tasks, some interacting closely with D-machine clock timing. Microinstruction

gating to several devices permits a single TRANSLANG instruction to accomplish some or all of the following actions:

a. test a condition.

b. set/reset a condition.

c. initiate an external operation.

d. add.

e. shift the result of an add.

f. store the result into several registers.

g. increment a counter.

h. complement a shift amount.

i. determine the successor microinstruction.

By judiciously composing his microprogram, a programmer may minimize execution time by taking advantage of microinstruction phase overlap and using highly parallel microcode.

The TRANSLANG assembler constructs an object program consisting of non-relocatable 56-bit microinstructions. TRANSLANG maintains a cross reference table that resolves label references during assembly. The object code created is stored on a disk and may be loaded into the micromemory of a D-machine using a special control word recognized by the operating system. Once loaded, the D-machine assumes the control structure dictated by the users microprogram.

V. AN/UYK-20 EMULATOR

A. EMULATION DESIGN

1. Functional Components

The architecture of the AN/UYK-20 emulator microprogram was developed using general guidelines provided by references and previous emulation experience on Burroughs D-machines [10]. The first decision incorporated in the emulator design was to integrate the entire emulation within one D-machine. Since the D-machines had the capacity to handle nearly 8K of microinstructions, no microprogramming capacity limitations were envisioned. The design objectives of a modularized, well-documented, structured microprogram could also be realized.

With the emulator entirely contained in one D-machine, several secondary benefits also existed. First, the emulator was more immune to hardware problems. If one D-machine was malfunctioning, the emulator could still be run on the alternate D-machine. Second, it was possible to have two AN/UYK-20 emulators resident in the system at one time. Not only would two emulators speed up testing of the individual emulation, but they would also permit their eventual use in a system configured for AN/UYK-20 multiprocessing. The third and final benefit of a totally integrated

emulator was recognized during the design of the input/output controller (IOC). Since the AN/UYK-20's IOC was capable of independent processing, an emulation of its IOC would not be possible within the same D-machine. An emulation of the IOC could be accomplished in a second D-machine, which would behave as an independent channel for the D-machine configured as the AN/UYK-20 processor. Although this emulation does not attempt to emulate the AN/UYK-20 IOC, the fact that a second D-machine exists makes its implementation a realistic extension to the project.

The emulator program organization was created following the basic guidelines of Ref. 17. The loader occupied the lowest section of M-memory with the emulator microcode following sequentially. Microprogram control passed from the loader to the emulator via the execution of a 'G' card which signified execution commencement.

The emulator microprogram was organized into three modules positioned such that forward address referencing would be minimized in the TRANSLANG assembler to save time and space. The utilities section was in the lowest portion of the emulator, since its routines were referenced frequently by the succeeding sections. Individual subroutines within the utility section were organized alphabetically.

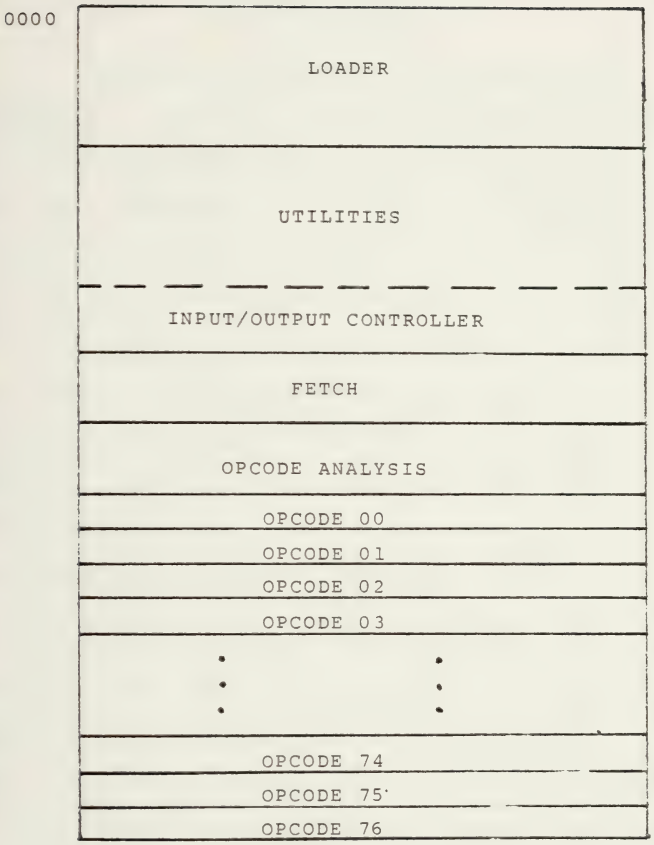
The instruction and memory fetch routines comprised the next module. These routines incorporated all fetch options available in the AN/UYK-20 emulator.

The opcode routines occupied the last section of the emulator. The opcodes were arranged numerically with further indexing determined by the remaining fields of the individual instruction. Figure V-1 shows the M-memory mapping of the AN/UYK-20 emulator layout. Appendix F contains the emulation program listing.

2. Main Memory Organization

Of primary importance in the emulation design was the logical organization of the Burroughs system's main memory. Since the AN/UYK-20 was a 16-bit word machine, it was decided that only the lower half-word of a Burroughs 32-bit word would be used by the emulator. This design restriction permitted the addressing structure of the AN/UYK-20 to be directly projected on the D-machine. A one-to-one correspondence between the memory address of the AN/UYK-20 and the Burroughs system was also achieved.

Since the number of high-speed registers available in the D-machine was small, a 1K portion of main memory was reserved for the AN/UYK-20 addressable register set, the page address registers, the temporary storage space, and the emulation buffers. The mapping of the AN/UYK-20 high-speed registers to main memory greatly simplified the microprogramming requirements of the emulation, but added considerable execution time overhead. Memory access times for the mapped registers were much slower than the actual transfer



AN/UYK-20 EMULATION ORGANIZATION

Figure V-1

rates of the AN/UYK-20 registers. Figure V-2 shows the complete main memory mapping of the emulation reserved storage area.

3. Emulation Program Status Word

Although the emulation duplicated all the AN/UYK-20 registers, the registers which comprised the program status word (PSW) possessed unique characteristics. Status register 1 was combined with the program address register to form a single 32-bit PSW. The emulator's PSW always co-existed in the A1 register of the D-machine and in main memory during execution of AN/UYK-20 programs. The fields of SR1 were modified to include certain emulator toggles, along with the normal condition bits (Figure V-3). The condition bits for DMA and non-destructive read only (NDRO) mode were removed from the SR1 since they were hardware features of the AN/UYK-20 that would not be emulated.

Status register 2, the remaining 16 bits of the AN-UYK-20's PSW, was not resident in any D-machine registers during emulation execution for two reasons: the emulation could not afford the luxury of 48 bits of reserved register space, and SR2 was less frequently referenced than SR1 and the P-register. Consequently, SR2 had to be read from its reserved location in main memory. The contents of the upper 16 bits of SR2's memory location also contained additional emulation toggles which were used by the debugger package (Figure V-3).

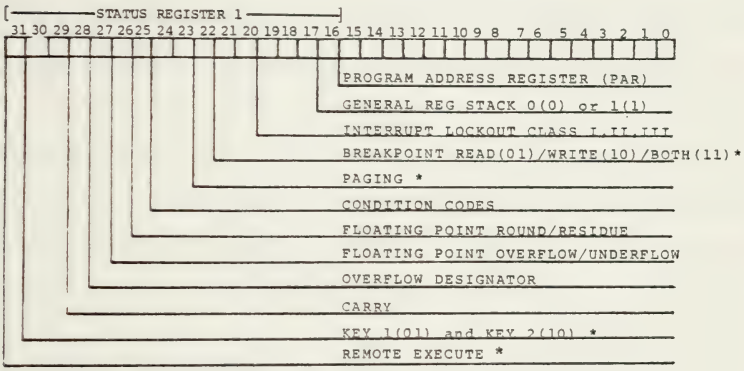
MAIN MEMORY MAPPINGS

DECIMAL ADDRESS	OCTAL ADDRESS	USE
0 - 15	0 - 17	GENERAL REGISTER STACK 1
16 - 31	20 - 37	GENERAL REGISTER STACK 2
32	40	PROGRAM STATUS WORD
33	41	BREAKPOINT REGISTER
34	42	STATUS REGISTER 2 (SR2)
35	43	NEXT LOAD ADDRESS
36	44	CLOCKTIME
37	45	REAL TIME CLOCK
38 - 43	46 - 53	WORKSPACE (TEMP STORAGE)
44 - 49	54 - 61	STACK (TEMP STORAGE STACK)
50 - 53	62 - 65	I/O COMMAND WORDS (IOCW)
54 - 119	66 - 167	UNUSED
120 - 121	170 - 171	HEX ADDRESS FOR INPUT
122 - 141	172 - 215	INPUT CARD BUFFER
142 - 152	216 - 230	UNUSED
153 - 185	231 - 271	OUTPUT PRINT BUFFER
186 - 205	272 - 315	CRT BUFFER
206 - 229	316 - 345	ERRORLIST
230 - 767	346 - 1377	UNUSED
768 - 1023	1400 - 1777	PAGE ADDRESS REGISTERS

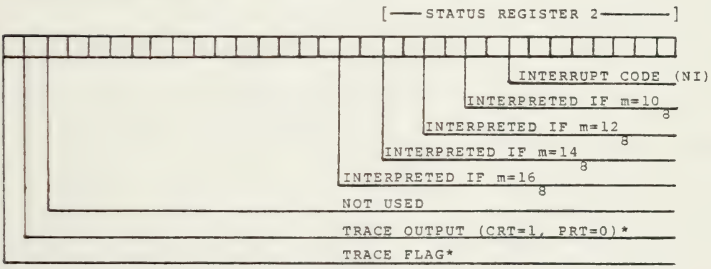
Figure V-2

AN/UYK-20 EMULATION
PROGRAM STATUS WORDS (PSW)

ACTIVE PSW: RESIDES IN LU REGISTER A1 (MEMORY ADDRESS 32)



INACTIVE PSW: RESIDES IN MEMORY ADDRESS 33



* FIELDS ADDED FOR USE BY THE EMULATOR

Figure V-3

4. D-Machine Registers

With only a limited number of high-speed registers available in the D-machine, the emulation design had to include a well-developed plan for their usage. Since only seven registers in the D-machine were 16 bits or longer, they were used exclusively for manipulating the AN/UYK-20 addresses, instructions, and data. The register environment had to be consistent throughout every execution cycle to allow utility routines to be called in the same manner by all opcode subroutines. The primary goal of the emulation was to use as few memory references as possible to conserve execution time and memory space. Judicious use of the existing registers was a necessity.

Several factors had to be considered before selecting a register for a particular function. The most important consideration was its flexibility within the D-machine. The B register, for example, was used as a general purpose register, since its contents could be gated through a masking filter prior to being utilized. A second factor was the type of operand it could contain. Double-word operands (32-bit) could only be stored in the 32-bit logic unit registers while 16-bit operands could also be stored in the memory control unit registers.

The A1 register contained the emulation's 32-bit program status word (PSW) at the commencement of the emulator program. It was not affected by individual instruction

microcode, except when incrementing the PAR, or when a particular instruction modified the PSW. Emulator toggles resident in SR1 could not be altered by an AN/UYK-20 instruction because their settings were independent of program execution.

The A3 register held the instruction word for the duration of its execution cycle. Each field of the instruction could be decoded and interpreted from the A3 register without having to retrieve it from memory. Once the instruction had been completely decoded, A3 was made available as a scratch pad register.

The A2 and B registers were used as scratch pad registers during each opcode execution cycle. In general, their contents were volatile, except when they were specifically documented in the the program listing. The contents of A2, for example, was not altered in the EMULIN subroutine, because of its use in the calling fetch routine. A2 and B were manipulated as either single or double-word operands during the arithmetic operations.

The only remaining logic unit register was the memory information register (MIR). MIR was used for storing information into memory and as a temporary storage location. Intermediate results were deposited in MIR during instruction execution and returned through the B register.

The base registers, BR1 and BR2, were used as storage for addresses and single-length operands, and for

temporary storage of intermediate results. In addition, all memory addressing in the emulation was accomplished using the lower 8 bits of BR2 and MAR (MAR2). These memory control unit registers had to be used carefully, because they required a sequence of several microinstructions to properly reference their contents.

B. LOADER

The loader incorporated into the AN/UYK-20 emulation provided a simple mechanism for loading AN/UYK-20 instructions into main memory (S-memory) of the Burroughs microprogramming system (Figure V-4). Its control word repertoire was flexible, allowing a variety of AN/UYK-20 program environments. Job control statements were included to execute and halt individual programs anywhere in S-memory.

The loader module consisted essentially of a scanner and a translator written in the microcode. Information was read into a 20-word buffer from cards or CRT input, then the buffer contents were scanned for control code consisting of one or more characters. Once these characters were interpreted, control was passed to the translator section which decoded the rest of the data in the buffer and performed the required function. The translator section consisted of a variety of routines that handled specific control words in the loader repertoire. The loader control statements, however, had to appear in a logical sequence (See Appendix A). All loader control statements are contained in Appendix B.

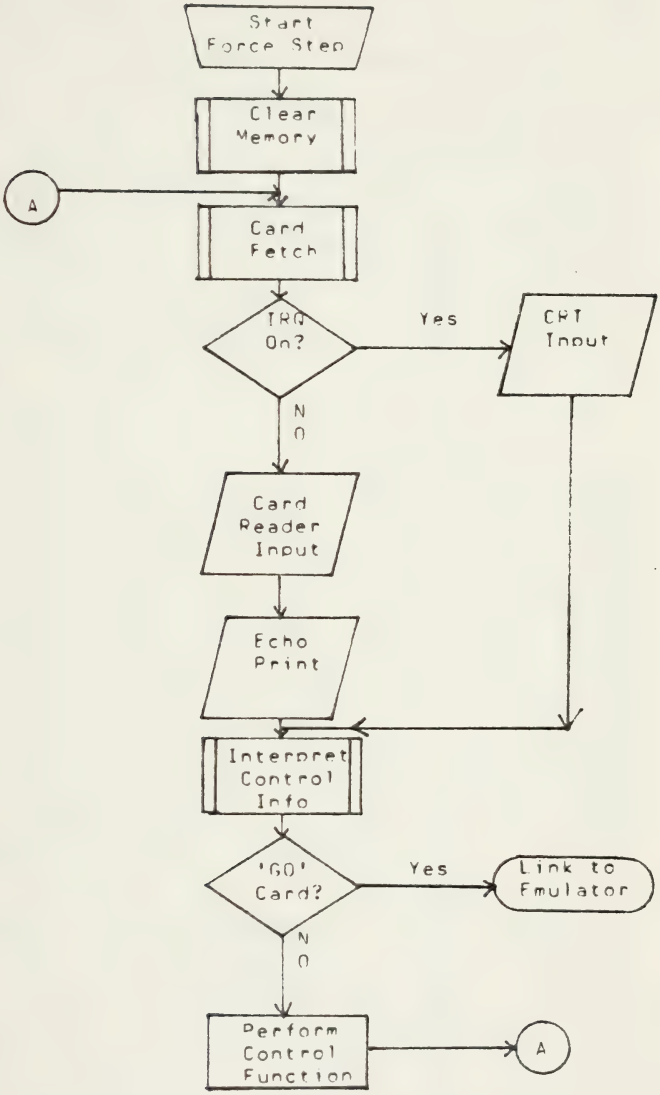

Start
Force Step
Clear
Memory
A
Card
Fetch
IRQ
On?
Yes
CRT
Input
N
O
Card
Reader
Input
Echo
Print
Interpret
Control
Info
'GO'
Card?
Yes
Link to
Emulator
N
O
Perform
Control
Function
A

Figure V-4

The actual interface pipeline between the loader and the emulator consisted of two small emulator utility routines which started and stopped the external clock of the Burroughs microprogramming system. These routines were inserted for emulation timing purposes, and provided a 'stop watch' for AN/UYK-20 programs. The time recorded (in milliseconds) by this 'stop watch' was placed in a reserved memory location in the emulation memory map, and could be read using either a trace control instruction ('T'), or a machine status control word ('M').

C. THE FETCH MODULE

In order to emulate AN/UYK-20 execution and memory referencing, fetch microcode was developed which incorporated memory addressing algorithms and instruction fetch routines. Since both data and instructions were equally accessible from the processor, the memory addressing scheme was closely linked to the instruction fetch concept. Data and instructions could be interspersed throughout memory, and proper program execution required that the program address register point to an instruction word.

The emulation used two routines for memory addressing, EMULIN for reading, and EMULOUT for writing (Figures V-5, V-6). These subroutines performed both paging and breakpoint checking, depending on toggles set in the program status word. Paging was incorporated into the emulation in order to gauge the execution overhead required in emulating

the AN/UYK-20's paging scheme. The paging scheme implemented in the emulator divided main memory into 256-word pages, instead of 1024-word pages used by the AN/UYK-20. Since the Burroughs D-machine was organized for 256-word pages, the microprogramming required for the paging was straightforward. The 256 page address registers resided in the emulator's memory mapping, each initialized to the page number corresponding to their relative address (0-255). The paging algorithm imitated the AN/UYK-20's method of page addressing, and included the setting of a page modification bit.

The breakpoint option was added to provide a method of debugging AN/UYK-20 programs, once the emulation was completed. EMULIN and EMULOUT tested toggles set in the PSW to determine if breakpoint read, write, or both was desired.

The memory addressing convention required all memory references from 1K to 64K to use EMULIN and EMULOUT. All memory references to the memory mapping area (0-1023) did not use these routines, but instead utilized absolute memory referencing microcode.

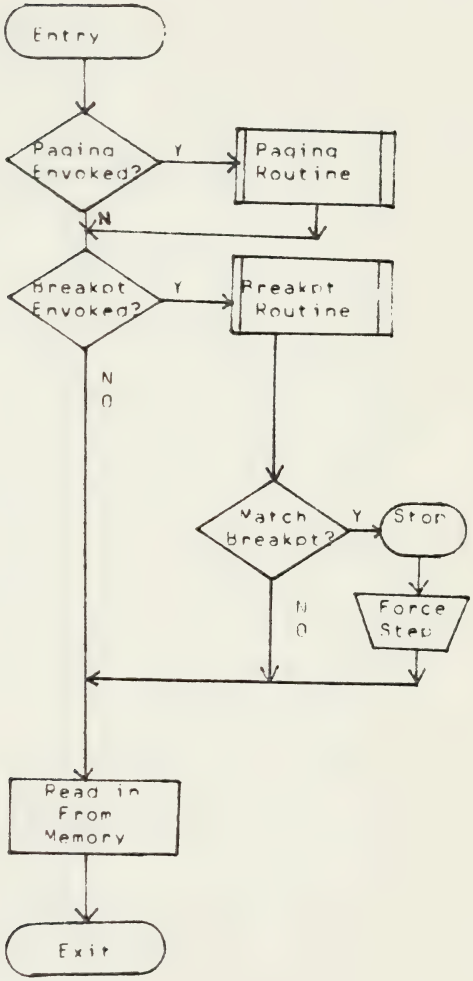

Entry
Paging Envoked?
Y
Paging Routine
N
Breakpt Envoked?
Y
Breakpt Routine
N O
Match Breakpt?
Y
Stop
N O
Force Step
Read in From Memory
Exit

Figure V-5

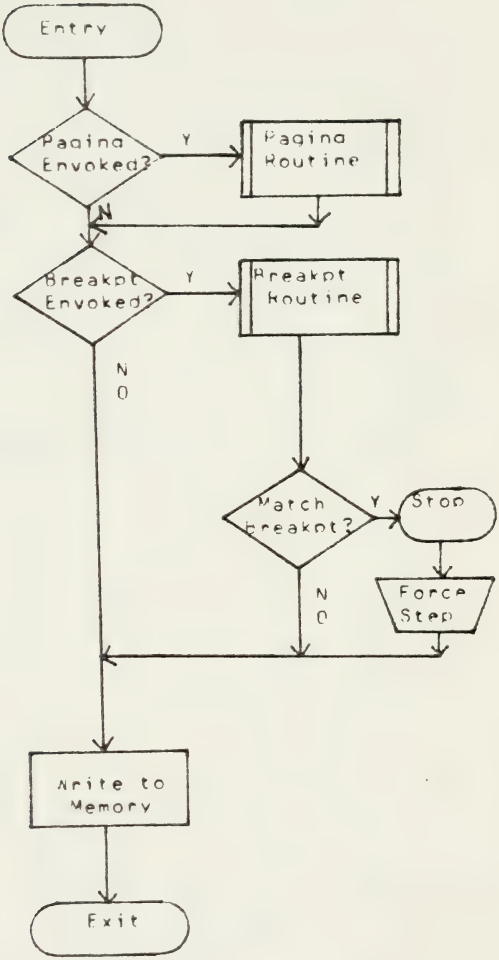

Entry
Paging Envoked?
Y
Paging Routine
N
Breakpt Envoked?
Y
Breakpt Routine
N O
Match Breakpt?
Y
Stop
Force Step
N O
Write to Memory
Exit

Figure V-6

The instruction fetch routine (IFETCH) retrieved instructions from main memory based on the contents of the program address register. Since IFETCH used EMULIN to read the instructions from memory, it could operate in a paging environment, with or without breakpoints (Figure V-7). IFETCH was also responsible for incrementing the PAR, and for testing the trace toggle prior to fetching an instruction. IFETCH was capable of retrieving any instruction word in the program address space (1024-65,535). In the event that the upper memory limit was reached, IFETCH would set the PAR to 1024 and continue execution. Trace toggle testing was inserted into IFETCH as part of the built-in debugging package. Since IFETCH was called prior to every instruction, it was the logical choice for placing a call to the debugger.

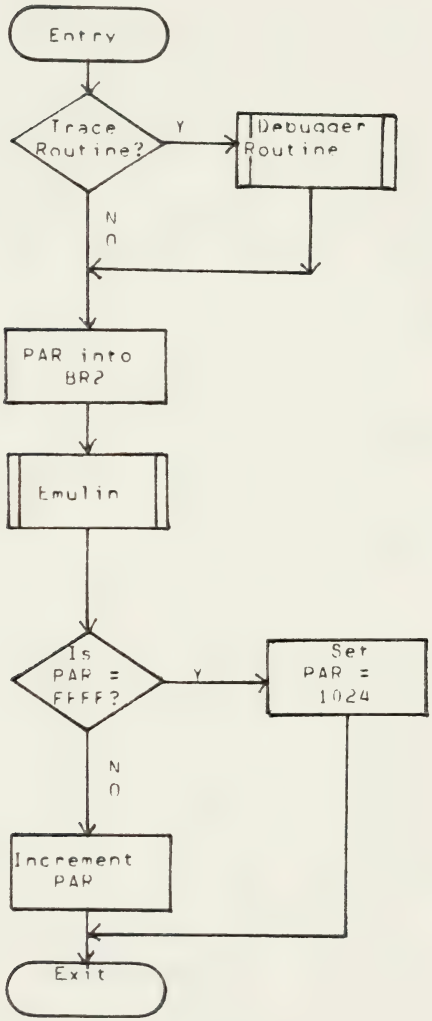

Entry
Trace Routine?
Y
Debugger Routine
N O
PAR into BR2
Emulin
Is PAR = FFFF?
Y
Set PAR = 1024
N O
Increment PAR
Exit

Figure V-7

D. OPCODE IMPLEMENTATION

All of the 205 individual instructions emulated were microprogrammed using an identical instruction decoding mechanism. The routines that performed the required operations were terminal nodes on a large tree network, whose root was the OPCODE routine. OPCODE fetched an instruction and isolated the operation code field. The binary value of the opcode field was an index to an operation jump table containing individual opcode M-memory addresses. Within each opcode routine was microcode which isolated the subfunction 'f' field of the instruction and used its value as an index to the next level of opcode analysis. Depending on the opcode, this last jump could identify which instruction was to be performed. If it did not, further analysis of the 'm' or 'a' fields provided the final index to the instruction. Instruction formats are described in Appendix C.

After the instruction had been executed, control passed back to the opcode routine, and then the execution cycle was repeated for the emulation of the next AN/UYK-20 instruction. The AN/UYK-20 programmer could cause this microprogram loop to be exited by either inserting an executive return instruction (03,0,a,00) which caused a 'priority interrupt' and halted program execution, or by coding an instruction that was not implemented, not assigned, or caused a division overflow. The last three cases caused an execution fault, while the former resulted in normal program termination (Figure V-8).

AN/UYK-20 EMULATION GENERAL STRUCTURE

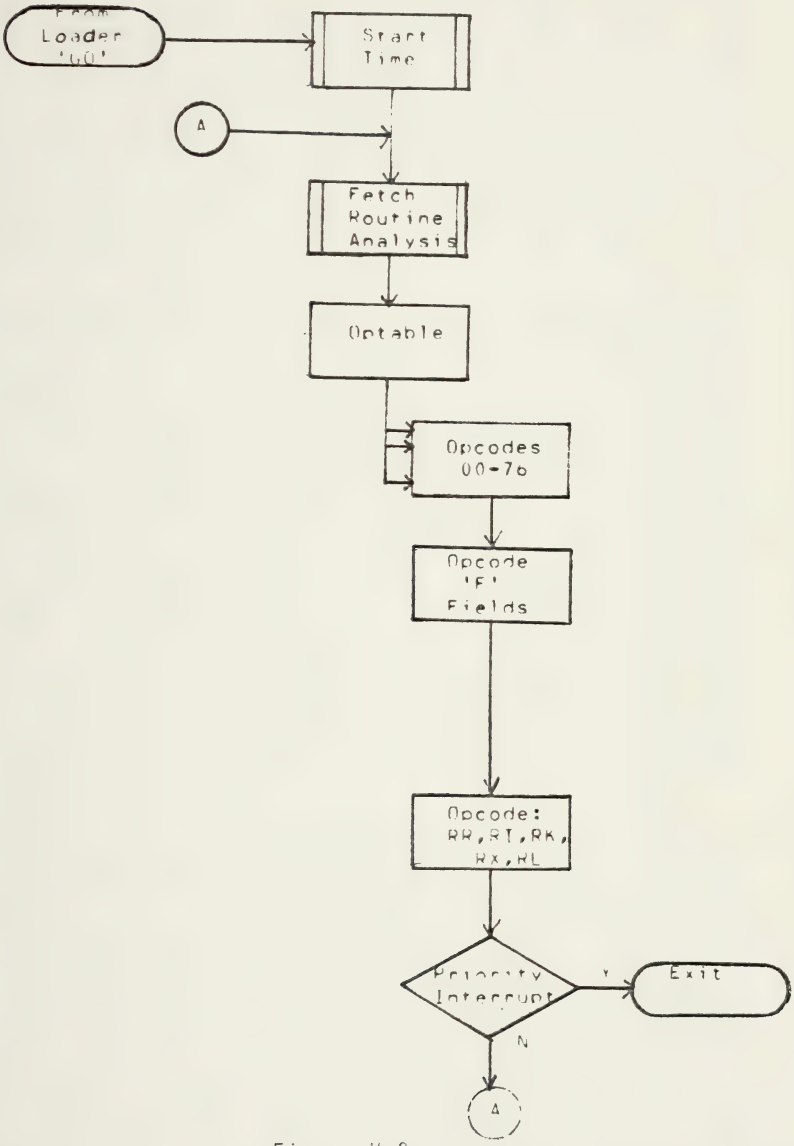


Figure V-8

The instruction fetch mode of OPCODE assumed that the PAR always pointed to an instruction word. Double-word instructions always had their 'y' field fetched after they were determined to be two words in length. Conditional double-word instructions performed their condition test before the 'y' field was fetched. If the test failed, the PAR was incremented by two prior to returning to OPCODE for continued execution.

The OPCODE routine was also written to accommodate the AN/UYK-20 'remote execute' instruction. When this operation was performed, a bit was set in the PSW which would indicate that one instruction out-of-line was being executed. For this operation, the current PSW was stored in memory, and the PAR was loaded with the address of the instruction to be executed. The OPCODE routine always checked the remote execute bit during an instruction cycle. If the bit was set, it would fetch the instruction indicated by the remote execute PAR, and restore the actual PAR, incremented by two, into the PSW.

Some of the opcode repertoire of the AN/UYK-20 was not emulated. Those instructions that were not emulated, however, retained slots in the opcode hierarchy for future inclusion. Any instruction not implemented by the emulator caused the machine to fault, and printed an error diagnostic on the selected output device ('NOT IMPLEMENTED - EXECUTION ENDS '). Similarly, locations were reserved for those instructions not assigned by the AN/UYK-20 were reserved

locations in the emulation . This permitted the emulator to be responsive to any future AN/UYK-20 hardware modifications. Whenever an instruction to be executed was not assigned, the emulator generated a fault interrupt and printed an error diagnostic on the selected output device ('FAULT INTERRUPT - EXECUTION ENDS').

E. UTILITIES

Although each emulated instruction performed different operations, each depended upon a common set of utility subroutines to accomplish their task. These subroutines varied in complexity, but each performed a function that contributed to successful instruction execution. A simple operation often required register addressing, condition code setting, and memory addressing, before the task was completed.

Utility subroutines were included in the emulation whenever feasible to simplify microprogramming and to alleviate programming redundancy. These subroutines were called using the successor command constructs available in TRANSLANG. Depending on the purpose of the utility routine, parameters were passed via D-machine register(s) or condition bit(s). This information was utilized by the utility in determining what operation was to be performed. In the carry subroutine, for example, a local condition bit was passed which indicated the appropriate condition code to be inserted into the PSW. A more complex example, the RX

format utility routine, required two parameters to be set by the instruction. Register A3 contained the instruction word and LC2 was set or cleared depending on whether or not byte formatting was required. The RX routine called other routines and could perform considerable processing before the final result, the effective operand address, was returned to the calling routine via the B register.

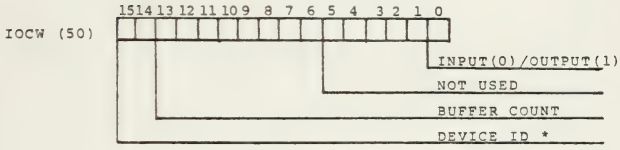
The utility section encompassed a number of routines which performed complex data manipulation. Arithmetic utilities for multiplication, division, and square root were accessed by nine separate AN/UYK-20 instructions. Indirection routines, which were called by the RX format routines, emulated the AN/UYK-20 cascaded addressing capabilities. A general purpose move subroutine permitted up to 256 cells of main memory to be moved from one location to another. This routine was used by the emulator's error diagnostic utilities, as well as the load and store multiple address register instructions.

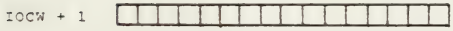
F. INPUT/OUTPUT CONTROLLER

Although the AN/UYK-20's IOC was not emulated, several of its design features were imitated in creating a general purpose input/output controller for the emulator. The input/output command instruction (35 RR) initiated the I/O sequence, and reserved several cells in the memory mapping for I/O control words (Figure V-9). These control words contained fields which indicated what peripheral device was

AN/UYK-20 EMULATION

INPUT/OUTPUT COMMAND WORDS





BUFFER ADDRESS
THE MICROCODE ALTERS THE IOCW + 1
DURING BUFFER TRANSFER

BUFFER SIZE
USED FOR CRT OUTPUT

NUMBER OF SECTIONS
USED FOR DISK TRANSFERS (NOT IMPLEMENTED)

* CODE: 00 - CRT
01 - PRT (OUTPUT)/CRD RDR (INPUT)
10 - DISK (NOT IMPLEMENTED)

Figure V-9

selected, the number of buffers that would be passed, whether input or output was desired, and where the buffer was located. Two additional words were reserved for control data required to interface with the disk and the CRT. The disk input/output microcode was not incorporated into the controller, but the framework was provided so the IOC could be readily inserted in the future.

The emulator's IOC section was located in the utility section of the emulation, and operated asynchronously with the Burroughs IOP. Since the IOC was collocated with the emulator in the same D-machine, it was not capable of independent processing. Once it was initiated, emulation execution waited until input/output was completed. The emulation had to use the Burroughs Common Language, since the IOP was microprogrammed to accept only this character set from other D-machines.

In order to transfer a 33-word line printer buffer to the IOP, the AN/UYK-20 emulator IOC had to use 66 AN/UYK-20 words. In order to send 20 words to the CRT, the IOC had to contruct a 40-word buffer. The COMPRES subroutine packed an AN/UYK-20 buffer of 66, 16-bit words into a 32-bit, 33-word buffer, suitable for output to the IOP.

Similarly, on input, a 20-word IOP buffer from the card reader or CRT expanded into a 40-word AN/UYK-20 buffer. In this case, the EXPAND subroutine split every packed 32-bit word of the IOP's buffer into two 16-bit words, suitable for processing by the emulator. These additional data transformations were required because of the buffering requirements of the Burroughs IOP.

VI. EMULATION TESTING

A. METHOD OF TESTING

The fundamental testing technique utilized throughout the development of the emulation consisted of three distinct phases: 1) each module was independently assembled and debugged until successful assembly was achieved, 2) each program segment was then tested for accuracy of the intended operation, and 3) the composite emulation program was tested under actual program conditions allowing for interaction among several modules. A debugging package, which was developed early in the emulation project, was the primary test vehicle for verifying emulation coding. Development and testing of the loader was, however, completed prior to the creation of the debugger.

Loader testing was accomplished by monitoring the control panel lights and synthetically halting program execution via the insertion of 'WAIT' statements, forcing the desired register to be transferred to 'MIR', and then re-examining the panel lights. This process was extremely tedious and time consuming but proved to be an invaluable tool for recognizing peculiar TRANSLANG constructs and microinstruction execution side effects.

Each module was subjected to an extensive desk checking process in order to reduce trivial assembly errors before running it on the machine. After a successful assembly had been achieved, the code was merged with previously existing assembled modules. The emulator was expanded as more modules were added, with utility routines being incorporated prior to the opcode programs.

Initially, the loader was designed, programmed, and thoroughly tested prior to the microprogramming of any AN/UYK-20 instructions or utilities. With the loader implemented, the emulation of the UYK-20 instruction repertoire could procede with a minimum of loader induced problems.

Independent opcodes and various utility routines were added to the previously assembled programs. The final emulation consisted of a total of 205 opcodes, a set of utility programs, and a workable loader.

In the next phase of testing, every module was subjected to a representative number of test cases which demonstrated how closely they compared to the documented AN/UYK-20 operation. Artificial environments were created to subject every opcode to a variety of situations. Whenever the operation of the opcode disagreed with the documented AN/UYK-20 operation, the opcode's function was thoroughly researched. The opcode's side effects were categorized and questions formulated. Often the answers could only be obtained from the Univac field engineer.

To illustrate the complexity of testing, an add instruction was tested with numerous operand combinations: two positive operands, two negative operands, one of each sign yielding a negative result, opposite signs producing a positive result, and opposite signs producing a zero sum. During each addition operation, the overflow, carry, and condition bits had to be monitored in status register 1 to verify their appropriate setting. This level of detail was achieved with all of the implemented opcodes in order to produce an efficient and accurate emulation.

Throughout the entire testing scenario, which composed 20% of the emulation project, the debugger routine (DUMPREG) provided the necessary information for examining opcode execution. A representative sample debugger output is provided in Appendix D.

DUMPREG permitted a snapshot of both AN/UYK-20 general register stacks, PSW, SR1, and SR2, in addition to the D-machine registers (A1, A2, A3, BR1, AMPCR, MIR) and the Burroughs external clock. DUMPREG possessed sufficient flexibility to be incorporated into the microcode at any point. In the final emulator, however, the debugger is user specified, and will dump the AN/UYK-20 emulator environment either at every instruction fetch and program stop, or when called by a loader control card.

B. SAMPLE TEST PROGRAMS

After subjecting the entire emulation to extensive testing, some representative test programs were developed to demonstrate the feasibility of the emulation and its capabilities and performance as compared to an actual AN/UYK-20. There were two programs which were selected because they incorporated numerous emulation features. The two programs were the solution of simultaneous linear equations by Cramer's rule and generation of prime numbers. It must be noted that streamlined program design was not emphasized but rather utilization of a variety of opcodes and features of the emulation.

The program for solving linear equations contained a total of 28 opcodes, requiring 28 opcode execution cycles and 43 instruction fetches. The program demonstrated all four fundamental mathematical operations, numerous store and load functions, a comparison test and a jump instruction. The capability of performing card reader input and output was added when the emulator IOC was completed.

The prime number program demonstrated 30 instructions which required 116 opcode execution cycles, and 122 instruction fetches. This program illustrated numerous comparison tests, looping structures, several jump instructions, a load multiple instruction, addition, and division. The test programs are included in Appendix E.

C. TEST RESULTS

The performance analysis of the test programs consisted primarily of running numerous test cases, examining the results for accuracy and computing the total time required to execute the emulated AN/UYK-20 programs. The timing of the programs was accomplished by using an external clock available on the IOP. The clock time provided a fairly close representation of the emulation execution time, but it cannot be considered a completely accurate measure since the emulation must interrogate the IOP to retrieve the external clock contents. Approximately 50 microseconds are used when sampling the clock.

The time requirements of the AN/UYK-20 program execution were hand-calculated by summing the published instruction execution times as presented in Ref. 21. The emulation performance ratio (EPR) was computed merely to give an approximate indication of the emulation performance. The EPR is the ratio of measured Burroughs emulation time to the calculated AN/UYK-20 execution time. Two EPR figures are recorded: one with paging implemented and one without. The paging EPR figure is significant only in that it indicates how much additional overhead must be incurred when emulating the paging mechanism of the AN/UYK-20. The required additional execution time was about 26%. Naturally, paging overhead is directly proportional to the number of instruction fetches or memory references performed during a program.

The results of program testing are as follows:

	CRAMER'S RULE	PRIME NUMBERS
Number of Opcodes	28	30
Number of Fetches	43	122
Execution Cycles	28	116
Time w/o paging	5000 usec	13000 usec
Time with Paging	8000 usec	17000 usec
AN/UYK-20 Time	78 usec	193 usec
EPR w/o Paging	64 :: 1	67 :: 1
EPR with Paging	103 :: 1	88 :: 1

These figures represent only approximate comparisons of the two machines. These computations provide an estimate of emulation characteristics. An effective EPR without paging was projected to be 65::1.

VII. SUMMARY AND RECOMMENDATIONS

A. EXPERIENCE WITH HARDWARE

The emulation project provided the unique opportunity of learning about two computer systems. The Burroughs D-machine demanded a detailed knowledge of hardware operation, as well as a thorough understanding of the microprogramming language, TRANSLANG. The computer architecture and processor capabilities of the AN/UYK-20 had to be investigated and then integrated into the control store memory of the Burrough's D-machine.

On several occasions, hardware malfunctions with the Burroughs equipment prevented normal system operation. The card reader was inoperable for several weeks, and the disk drive unit had to be repaired several times. During the final three weeks of project development, one D-machine ceased to function properly. This restricted emulation testing to the remaining D-machine.

Although these hardware difficulties impeded normal progress of the project, they did not prevent the emulation from successfully being completed. If a hardware problem prevented emulation testing or debugging, other modules were designed and testing was postponed. This permitted continual emulation development regardless of the hardware

status. In addition, considerable time and effort was invested in trouble-shooting hardware malfunctions, so they could be isolated, diagnosed, and repaired.

B. LESSONS LEARNED

The emulation project brought together many different computer science techniques and disciplines which will be useful in future computer science endeavors. A great deal of experience in both computer architecture and operation was gained in two different types of computers. Microprogramming provided new insight into computer design and revealed many potential applications for programmable control store machines.

Since emulations normally require a large development effort, this project had to incorporate judicious system design and project management principles in order for it to be completed successfully. Careful monitoring of critical stages of the emulation, and coordination of the programming effort to meet scheduled requirements, was necessary throughout this research. The small programming team concept proved to work extremely well in this project.

Finally, several software practices were strictly followed that proved to be invaluable in constructing the emulation. First, modular programming succeeded in partitioning the emulation into discrete modules which could be designed, coded, tested, and implemented individually. The

emulation was constructed in segments, using the previously verified modules as a test bed for the new modules being built. Structured programming and prolific documentation were useful in developing each microprogram routine because they permitted each team member to understand the function of the program and how it could be used.

C. EMULATION PROBLEMS

The most significant problem of the emulation was not having had any experience with an AN/UYK-20 and not having anyone readily available with prior AN/UYK-20 experience. This lack of knowledge created some anxiety when attempting to analyze the ramifications of individual opcodes.

The idiosyncrasies of certain opcodes were not made self-evident by the AN/UYK-20 software manuals. Consequently, numerous code modules were redesigned when more detailed information was provided by a UNIVAC field engineer. This proved to be very time consuming, inefficient, and frustrating.

Another emulation difficulty was the lack of working registers in the host machine as compared with the target machine. While the AN/UYK-20 has either 16 or 32 general registers, depending on whether the second stack option is incorporated, the D-machine has effectively only seven workable registers containing 16 bits or more. Therefore, all AN/UYK-20 registers had to be mapped into S-memory which

created much longer register read/write times. This created significant register manipulation problems which in some cases required main memory references for instructions intended to be strictly register-to-register operations. Consequently, increased execution times resulted in a higher emulation performance ratio (EPR), decreasing the overall emulation performance.

D. RESULTS

Emulating the AN/UYK-20 on a Burroughs D-machine required a considerable amount of preparation and planning before any results were realized. An in depth analysis of each computer's architecture and operating characteristics was conducted to insure that an emulation was feasible in the allotted time period. From the inception of the project, the goal of the emulation was to implement a standard AN/UYK-20 processor. This decision was based on the capability of the D-machine and an estimate of how much time would be involved in developing, debugging, and testing the final product. Although this goal was achieved, it was felt more time could have been devoted to testing and verifying emulation operation.

The AN/UYK-20 emulation was a highly complex microprogramming project involving numerous data structures and transfer protocols. A total of 205 AN/UYK-20 instructions were emulated out of nearly 290 instructions in the repertoire (including all IOC, math pac, and clock interrupt

opcodes). AN/UYK-20 diagnostic programs and user programs will establish the validity of the emulator's construction.

E. RECOMMENDATIONS AND FOLLOW-ON TOPICS

Although 205 opcodes have been implemented, tested, and developed into a working emulation, there are still many challenging avenues to pursue in creating the complete emulation package. First, testing is a continuous process and should be performed in conjunction with code optimization. A comprehensive code optimization effort could improve the EPR without sacrificing code readability.

Second, the floating point and 'math pac' options could be implemented. The addition of floating point arithmetic, trigonometric, and hyperbolic functions would significantly strengthen the scientific capabilities of the emulation. This would be an extremely strenuous undertaking but would permit more tactical data applications of the emulation, increasing its value to the Navy.

One area which could be examined is to perform a comprehensive timing analysis between an AN/UYK-20 and the emulation. This could consist of collecting numerous bench mark programs from Navy AN/UYK-20 installations, where performance data could be accurately obtained. These programs could then be run on the emulator, and analyzed with the bench mark results. The feasibility of replacing or substituting emulation host machines for target machines could be

addressed and supported by the timing analysis study.

An emulation of the AN/UYK-20 input/output controller could be incorporated into the emulation without serious difficulty. Since a second D-machine is available, the IOC channel processor instructions could be emulated and inserted into that D-machine's control memory. The independent processing characteristic of the AN/UYK-20 IOC would be fulfilled using this arrangement.

Finally, when the Burroughs system is linked with the computer science department's PDP 11/50, the AN/UYK-20 emulator could be connected to that system's peripheral resources. This would allow future incorporation of an AN/UYK-20 ULTRA assembler and a CMS-2M compiler into the PDP 11/50 system which could then produce machine language object code files for the AN/UYK-20 emulator to execute.

APPENDIX A. AN/UYK-20 EMULATOR USER'S MANUAL

This description is designed to provide sufficient information to operate the AN/UYK-20 emulation program. It is assumed that the informal Burroughs D-machine manual in the Burroughs laboratory will supply adequate power-on instructions and solve any hardware operating difficulties which may arise.

The procedure for utilizing the AN/UYK-20 emulator can be divided into four phases:

1) selection of the necessary loader control cards (JCL).
2) selection of the AN/UYK-20 program instruction set.
3) implementation of phases one and two into the required card or CRT format.
4) actual hardware implementation on the Burroughs D-machine resulting in program execution.

Phase one can be achieved by selecting the desired loader control cards described in Appendix B. The card format is identical to the CRT format, except that the CRT requires the user to <carriage return> at the end of every line of input data.

Phase two is accomplished by creating the AN/UYK-20 program from a subset of the 205 emulated instructions.

Phase three consists of keypunching the desired JCL and AN/UYK-20 program in the format illustrated in Appendix C.

The resulting job deck will typically be assembled as follows:

```
?SMLOAD TED/UYK20-OBJECT      % loads the emulator into
                                micromemory
L  00004                      % load the program into
                                page 4

C  00206 FAULT INTERRUPT-EXECUTION ENDS "

C  00214 NOT IMPLEMENTED-EXECUTION ENDS "

C  00222 DIVIDE OVERFLOW - FAULT ENTERED"

************************
other desired JCL
************************

AN/UYK-20 Program

    03,0,a,00                 % priority interrupt
    (a = 00-17 octal)           (mandatory card)

G                             % commence program
                                execution

M (or M1)                     % machine status (reg. dump
                                at termination)

E                             % end job card
```

Finally, phase four consists of the entire program deck being loaded into the card reader. It is assumed that the IOP is at address 0015 hex, the selected interpreter is at address 0549 hex, the line printer is 'READY', and the IRQ switch is 'off'.

The IRQ switch is a three-way toggle switch mounted on the right side of the interpreter. In the up position, IRQ is 'on', horizontally it is 'off', and the downward position is used for external functions.

If either the interpreter or the IOP is not at the proper starting address, clear them by depressing the 'CLEAR' button on each unit. If this procedure does not remedy the situation, consult the user's manual in the laboratory.

Upon reading the ?SMLOAD card, the system will load the AN/UYK-20 emulator object program into the micromemory of the selected interpreter. The program address counter (on the interpreter) will be at the beginning of the emulation program, address 0000 hex. At this point, the user can select CRT input and output by placing the IRQ switch to the 'on' (upward) position. If CRT input is not desired, leave the IRQ switch in the 'off' (horizontal) position. Next, force step the interpreter by momentarily depressing the FST button (uppermost push button on side panel of the interpreter). Do not hold in the FST button. This may cause some undesirable side effects.

After depressing the FST button, the AN/UYK-20 program is loaded into S-memory. If the input is expected from the CRT, the user must enter his program from the console. Otherwise, the emulator will request cards from the card reader. Once the program has been loaded and the 'G' card read, program execution begins.

After successful program termination, the program returns to the loader and asks for more input. At this point, the user may terminate his job, or start another load sequence. It should be remembered that the AN/UYK-20 emulator has been designed for monoprogramming execution. It executes one program at a time, but can execute any number of programs in sequential order if desired. When the job is terminated, the D-machine returns to its starting address (0000 hex) and awaits further processing. When the user returns to the start address, the system is effectively 'master cleared' since S-memory will be cleared prior to executing any further jobs. It is not necessary, however, to use the ?SMLOAD card for additional programs or program re-runs because the emulator object code still resides in micromemory.

If the results of program execution are not as anticipated, use of either a 'T' or 'T1' option (trace card) is recommended to provide the user with a fetch-by-fetch program trace with the output to the printer or CRT respectively.

APPENDIX B. LOADER CONTROL CARD FORMATS

CARD COLUMNS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16 - 20
B	R W B		decimal			addr									
C			decimal			addr		character string followed by a "							
D			decimal			addr									data
E															
G			decimal			addr									
I	1 2					req#									data
L			dec. page num												
M	1														
P															
R			dec. number												
S	1 2														
T	1														

Note: All numeric fields are in decimal.

LOADER CONTROL CARD DESCRIPTION

Control Card Identitifier		Description
B	---	The 'B' card is used to implement the breakpoint feature of the emulation. This features allows the user to specify a decimal address in columns 4-8. Column 2 must contain a R or W to breakpoint on a read or write operation respectively. The default condition is breakpoint on both read and write.
C	---	The 'C' card is used to insert character strings into memory. The character string starts in column 9 and continues until terminated by a quote symbol (") or the string reaches column 80. The decimal address where the character string will be written is given in columns 4-8. A blank or zero address field causes the character string to be inserted at the current load address.
D	---	The 'D' card is used to store decimal data from columns 16-20 into the memory address found in columns 4-8. A blank or zero address field causes the data to be inserted at the currrent load address.
E	---	The 'E' card is used to indicate the end-of-job and therefore is a mandatory control card for job separation.
G	---	The 'G' card or 'GO' card is used to start execution. The starting decimal address is in columns 4-8. The default value of a blank or zero address field will cause program execution to start address 01024.

I --- The 'I' card or set index register card is used to store decimal data from columns 16-20 into the register designated in columns 7-8, into the general register stack (1 or 2) specified in column 2.

L --- The 'L' or load card is used to partition memory into 256, 256-word (32-bit) pages and to load the program into the decimal page as referenced by columns 4-8. Pages 0-3 should not be used because they contain the required emulation register mapping and established workspace. The 'L' is a required control card, and it is recommended that it be the first JCL card.

M --- The 'M' card or machine status card provides a register dump, whereever it is inserted. It is normally placed between the 'G' and 'E' cards in the JCL deck. If a 1 is placed in column 2, the machine status dump will be sent to the CRT. This dump will contain the current value all AN/UYK-20 general registers, the PSW, SR2, next instruction address, the breakpoint address, and clocktime.

P --- The 'P' card is used to implement paging.

R --- The 'R' or reserve space card is used to reserve memory space as specified decimally in columns 4-8.

S --- The 'S' card is used to simulate setting of the program stop switches (1 and 2) on the AN/UYK-20 maintenance panel. Column 2 must contain a 1 or a 2. Two cards are required if both switches are to be set.

T --- The 'T' or trace card provides a fetch-by-fetch program trace, dumping the entire machine status on every fetch cycle. If a 1 is in column 2, the trace will be displayed on the CRT. This card is recommended for debugging programs.

APPENDIX C. AN/UYK-20 EMULATOR INSTRUCTION FORMAT

Format	Card Columns														
	5	6	7	8	9	10	11	12	13	14					
RR, RL, RI Type 2	opcode	,	'f'	,		' a '		,	' m '						
	5	6	7	8	9	10	11	12							
RI Type 1	opcode	,	'f'	,		' d '									
	5	6	7	8	9	10	11	12	13	14	15	16 - 20			
RK, RX	opcode	,	'f'	,		' a '	,		' m '	,		' y '			

Notes:

1) All fields are in octal with the exceptions of addresses and the 'y' field which are decimal
2) All fields must be zero-filled
3) The following field restrictions must be followed:

Field	Range	Base
Opcode	00-76	octal
'f'	0-3	octal
'a'	00-17	octal
'm'	00-17	octal
'd'	000-377	octal
'y'	00000-65535	decimal
addr	00000-65535	decimal

APPENDIX D. SAMPLE DEBUGGER OUTPUT

This appendix provides a sample program output illustrating the Trace option. The debugger is called at the beginning of every instruction fetch, and at the termination of the program. Space has been provided for dumping 48 memory addresses including emulated AN/UYK-20 registers and D-machine registers. Each output line consists of eight regions printed in hexadecimal with a total of eight hex digits (32 bits) per region.

The listing is annotated to indicate the identity of each output cell. A summary of cell definitions (numbering from left to right, top to bottom) follows:

DECIMAL ADDRESS	DESCRIPTION
0 - 15	General Register Stack 1
16 - 31	General Register Stack 2
32	Program Status Word (PSW)
33	Breakpoint Register (BREAKPT)
34	Status Register 2 (SR2)
35	NEXTINSTR (next load address)
36	Clocktime
37	Real-Time Clock (RTC)
38	AMPCR
39	A1
40	A2
41	A3
42	MIR
43	BR1
44 - 47	Unused

```
L   00004
C   00236 FAULT INTERRUPT-EXECUTION ENDS '
C   00214 NOT IMPLEMENTED-EXECUTION ENDS '
C   00222 DIVIDE OVERFLOW - FAULT ENTERED'
T                                   ENABLE DEBUGGER TRACE TO PRT
P                                   ENABLE PAGING
B   01025                           BREAKPOINT (R/W) AT ADDRESS 1025 DECIMAL
I1     01        00015
I1     02        00010
     02,0,02,01                     MAKE REG 2 NEGATIVE
     22,0,01,02                     ADD REG 2 TO REG 1 AND STORE RESULT IN REG 1
     03,0,00,00                     EXECUTIVE RETURN (HALT)
G                                   COMMENCE EXECUTION
00000000   00000000A   00000000   00000000   00000000   00000000   00000000
00000000   00000000   00000000   00000000   00000000   00000000   00000000 (15)
00000000   00000000   00000000   00000000   00000000   00000000   00000000
00000000   00000000   00000000   00000000   00000000   00000000   00000000 (31)
00700400   80000000   00000403   00000000   00000000   0000063A   00700400
00000672   30000000   00000002   00000000   00000000   00000000   00000000 (47)


00000000   00000000   00000000   00000000   00000000   00000000   00000000
00000000   00000000   00000000   00000000   00000000   00000000   00000000 GEN REG STACK 1
00000000   00000000   00000000   00000000   00000000   00000000   00000000
00000000   00000000   00000000   00000000   00000000   00000000   00000000 GEN REG STACK 2
00700400   80000000   00000403   00000000   00000000   0000063A   01F00401
00000672   00000821   00000002   00000000   00000000   00000000   00000000


00000000   00000035   00000000   00000000   00000000   00000000   00000000
00000000   00000000   00000000   00000000   00000000   00000000   00000000
00000000   00000000   00000000   00000000   00000000   00000000   00000000
00000000   00000000   00000000   00000000   00000000   00000000   00000000
00700400   80000000   00000403   00000000   00000000   0000063A   10F00402
00000672   00014812   00000005   00000002   00000000   00000000   00000000
PSW        BREAKPT    SR2        NEXTINSTR  CLOCKTIME  RTC        AMPCR      A1

00000404   00000035   00000000   00000000   00000000   00000000   00000000
00000000   00000000   00000000   00000000   00000000   00000000   00000000
00000000   00000000   00000000   00000000   00000000   00000000   00000000
00000000   00000000   00000000   00000000   00000000   00000000   00000000
00700400   80000000   00003E2C   00000032   00000000   00000000   10F00403
00000000   00000000   00000000   00000000   00000000   00000000   00000000
A2         A3         MIR        BRI        [<------- 44 - 47 ------->]

E                                   END OF JOB
```

APPENDIX E. SAMPLE TEST PROGRAMS

The programs listed in this appendix were designed to illustrate how the AN/UYK-20 emulator can be used for developing programs. The first two programs presented are the generation of prime numbers and the solution of simultaneous linear equations by Cramer's Rule. They depict several AN/UYK-20 control structures such as looping, iteration, and condition code testing as well as numerous load, store, and arithmetic operations. The last program performs the Cramer's Rule algorithm and demonstrates input/output routines.

```
L  00034                          PRIME NUMBER GENERATOR
N                                 PRIME NUMBERS RETURNED IN GEN. REGISTERS
P                                 THIS CARD WILL ENVOKE PAGING
C  00206 FAULT INTERRUPT-EXECUTION ENDS *
C  00214 NOT IMPLEMENTED-EXECUTION ENDS *
C  00222 DIVIDE OVERFLOW - FAULT ENTERED*
    63,0,04,01                    LITERAL LOAD, 1 INTO REG 4 (1ST PRIME)
    63,0,05,02                    LITERAL LOAD, 2 INTO REG 5 (2ND PRIME)
    63,0,06,02                    LITERAL LOAD, 2 INTO REG 6 (COUNTER)
    63,0,07,05                    LITERAL LOAD, NO. OF PRIMES INTO REG 7
    11,3,04,06,02048              STORE  A 1 INTO ADDR 2048
    11,3,05,00,02049              STORE  A 2 INTO ADDR 2049
    24,0,07,04                    COMPARE N (NO. OF PRIMES) & 1
    44,1,026                      LOCAL JUMP, IF EQUAL
    24,0,07,05                    COMPARE N AND 2
    44,1,023                      LOCAL JUMP, IF EQUAL
    63,0,11,02                    LITERAL LOAD, 2 INTO REG 11 (CANDIDATE)
    62,2,11,01                    LITERAL ADD, INCREMENT CANDIDATE
    63,0,13,02                    LITERAL LOAD, 2 INTO REG 10 (TRIAL DIVIS)
    01,0,13,11                    LOAD REG, CANDIDATE INTO REG 13
    01,0,12,00                    LOAD REG 12 WITH C
    27,0,12,10                    DIVIDE REG, CANDIDATE BY TRIAL DIVISOR
    24,0,09,12                    COMPARE, REMAINDER WITH 0
    44,1,371                      JUMP IF EQUAL, NON-PRIME, GET NEXT CAND.
    62,2,13,01                    LITERAL ADD, INCREMENT TRIAL DIVISOR
    24,0,10,11                    COMPARE DIVISOR WITH CANDIDATE
    47,1,370                      JUMP IF LESS THAN, TRY NEW DIVISOR
    11,3,11,06,02048              STORE FOUND PRIME INTO MEMORY
    62,2,05,01                    LITERAL ADD, INCREMENT PRIME COUNTER
    24,0,05,07                    COMPARE, PRIME COUNTER WITH N
    47,1,351                      JUMP IF LESS THAN, GET NEXT PRIME CAND.
    03,3,03,04,02048              LOAD MULT, PRIMES FROM MEMORY INTO REGS
    03,0,17,00                    PRIORITY INTERRUPT, HALT EXECUTION
    63,0,01,02                    LIT LOAD, FORCE FEED 2 INTO REG 01
    63,0,00,01                    LIT LOAD, FORCE FEED 1 INTO REG 00
    03,0,17,00                    PRIORITY INTERRUPT, HALT EXECUTION
G                                 GO, COMMENCE EXCUTION
M                                 MACHINE STATUS TO THE PRINTER
00000001  00000002  00000003  00000005  00000007  00000002  00000005  00000005
00000007  00000007  00000001  00000001  00000000  00000000  00000000  00000420
00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
00000400  00000000  00000000  00000422  00000018  00000000  00000161  00000000
00000030  00000000  00000000  00000005  00000000  00000000  00000000  00000000

E                                 END OF JOB
```

```
L   00034                          LINEAR EQUATIONS BY CRAMER'S RULE.
I1                                 X IS RETURNED IN REG 01, Y IN REG 05
C   00206 FAULT INTERRUPT-EXECUTION ENDS '
C   00214 NOT IMPLEMENTED-EXECUTION ENDS '
C   00222 DIVIDE OVERFLOW - FAULT ENTERED'
D   02048        00030             BOTH EQUATIONS MUST BE IN STD FORM
D   02049        00030             A (Y) PLUS B (X) EQUALS C
D   02050        00003             Y COEFF OF 1ST EQN
D   02051        65531             X COEFF OF 1ST EQN
D   02052        00019             CONSTANT OF 1ST EQN
D   02053        00002             Y COEFF OF 2ND EQN
D   02054        65532             X COEFF OF 2ND EQN
D   02055        00016             CONSTANT OF 2ND EQN
    01,3,01,00,02050               STORE X COEFF, EQN 1, INTO REG 01
    01,3,03,00,02051               STORE Y COEFF, EQN 1, INTO REG 03
    01,3,05,00,02052               STORE CONSTANT, EQN 1, INTO REG 05
    01,3,11,00,02053               STORE Y COEFF, EQN 2, INTO REG 11
    01,3,13,00,02054               STORE X COEFF, EQN 2, INTO REG 13
    01,3,15,00,02055               STORE CONSTANT, EQN 2, INTO REG 15
    26,0,00,13                     MULT LEFT DIAG OF DENOM, RESULT IN 00/01
    26,0,10,03                     MULT RT.  DIAG OF DENOM, RESULT IN 10/11
    21,0,00,10                     SUBTRACT DOUBLE, RESULT OF DENOM IN 00/01
    25,3,00,17,02048               COMPARE DENOM WITH ZERO
    44,1,052                       JUMP, IF EQUAL (ZERO DETERMINANT)
    12,3,03,06,00014               STORE DOUBLE, DENOM INTO 16/17
    01,2,03,00,00030               LOAD 00 INTO REG 00
    01,3,01,00,02050               RESTORE Y COEFF, EQN 1, INTO REG 01
    01,3,03,00,02051               RESTORE X COEFF, EQN 1, INTO REG 03
    01,3,05,00,02052               RESTORE CONSTANT OF EQN 1 IN REG 05
    01,3,11,00,02053               RESTORE Y COEFF OF EQN 2 IN REG 11
    01,3,13,00,02054               RESTORE X COEFF OF EQN 2 IN REG 13
    01,3,15,00,02055               RESTORE CONSTANT OF EQN 2 IN REG 15
    26,0,03,15                     X NUM LEFT DIAG MULT, RESULT IN 00/01
    26,0,13,05                     X NUM RT.  DIAG MULT, RESULT IN 10/11
    21,0,03,10                     X NUM RESULT IN REG 00/01
    27,0,00,17                     X RESIDES IN REG 01
    26,0,04,13                     Y NUM LEFT DIAG MULT, RESULT REG 04/05
    26,0,14,03                     Y NUM RT.  DIAG MULT, RESULT REG 14/15
    21,0,04,14                     DOUBLE SUBTRACT (EVAL NUM), RESULT 04/05
    27,0,04,17                     DIVIDE NUM/DENOM, Y RESIDES IN 05
    03,0,17,00                     PRIORITY INTERRUPT, HALT EXECUTION
G                                  GO, COMMENCE EXECUTION
H                                  MACHINE STATUS TO THE PRINTER
00000000   00300FFB   00300000   00000FFB   00000000   0000FFFE   00000000   00000000
00000000   00000026   00300000   00000FFC   00000FFF   0000FFB0   0000FFFF   00000420
00000000   00300000   00300000   00000030   00000000   00000000   00000000   00000000
00000000   00300000   00300000   00000030   00000030   00000000   00000000   00000000
00000400   00300000   00300000   00000428   00000005   00000000   00000161   00000000
00000030   00300050   00300000   00000034   00000000   00000000   00000000   00000000

F                                  END OF JOB
```

```
P                                   ****LINEAR EQUATIONS BY CRAMER:S RULE.****
L   00034
C   00236 FAULT INTERRUPT-EXECUTION ENDS '
C   00214 NOT IMPLEMENTED-EXECUTION ENDS '
C   00222 DIVIDE OVERFLOW - FAULT ENTERED'
I2      30        17654            IOCW FOR 20 BUFFERS FROM CARD READER
I2      31        04076
I2      32        16758            IOCW FOR 6 BUFFERS FROM CARD READER
I2      03        05120
I2      04        16735            OUTPUT IOCW FOR 5 BUFFERS TO PRT
I2      35        08076
I2      36        17655            OUTPUT 20 BUFFERS TO PRT
I2      37        10240
I2      08        12356            REGISTER CONTAINING TWO BCL SPACE CHARACTERS
I2      11        00040
I2      15        00030
D   02048         00030            BOTH EQUATIONS MUST BE IN STD FORM
D   02049         00030            A (Y) PLUS B (X) EQUALS C
D   02050         00031            Y COEFF OF 1ST EQN
D   02051         00031            X COEFF OF 1ST EQN
D   02052         00033            CONSTANT OF 1ST EQN
D   02053         00031            Y COEFF OF 2ND EQN
D   02054         65535            X COEFF OF 2ND EQN
D   02055         65535            CONSTANT OF 2ND EQN
     63,0,15,01                    LOAD 1 INTO REG. 13
     03,0,13,05                    LOAD SR1 WITH REG 13
     12,1,00,17                    STORE DOUBLE INTO IOCW
     35,0,03,00                    INPUT 14 BUFFERS FROM CARD READER
     12,1,02,17                    STORE DOUBLE INTO IOCW
     35,0,03,00                    INPUT 6 BUFFERS FROM THE CARD READER
     01,2,12,00,01320              INITIALIZE BUFFER WORDS COUNT (20 PRT BUFFERS)
     01,2,11,00,10240              INITIALIZE OUTPUT BUFFER ADDR
     15,1,10,11                    STORE (RA) INTO Y* AND INDEX BY 1 (STORE SPACES)
     02,0,12,11                    DECREMENT BUFFER COUNTER
     45,1,375                      COUNTER EQUAL ZERO
     01,2,14,00,04076              INITIALIZE BUFFER ADDRESS OF INPUT
     01,2,15,00,00020              BUFFER COUNT
     01,2,11,00,10240              INTIALIZE DESTINATION BUFFER ADDR
     05,1,15,14                    LOAD AND INDEX BY 1
     15,1,15,11                    STORE RA INTO Y* AND INDEX Y*
     02,0,13,11                    DECREMENT COUNTER
     45,1,374                      COUNTER EQUAL ZERO
     22,2,11,00,00026              ADD 26 TO BUFFER ADDR
     01,2,13,00,00040              RESET  BUFFER WORD COUNT
     02,0,15,11                    DECREMENT NEXT COUNTER (NO. OF BUFFERS)
     45,1,356                      COUNTER EQUAL ZERO
     12,1,05,17                    STORE DOUBLE INTO IOCW
     35,0,03,00                    OUTPUT 14 BUFFERS TO PRT
```

```
63,0,15,01          ***RE-ENTRY POINT FOR REPETITIVE EXECUTION***
03,0,13,05             LOAD SR1 WITH A 1 FROM REG 13
01,2,11,00,08096       INITIALIZE NEXT OUTPUT BUFFER ADDRESS
01,2,12,00,00330       INITIALIZE NEXT BUFFER WORD COUNT (5 PRT BUFFERS)
01,2,13,00,00035       BUFFER COUNT
01,2,14,00,05120       INITIALIZE  BUFFER ADDR OF SECOND INPUT BUFFER
15,1,10,11             STORE (RA) INTO Y* AND INDEX BY 1 (SKIPE SPACES)
02,0,12,11             DECREMENT BUFFER COUNTER
45,1,375               COUNTER EQUAL ZERO
01,2,11,00,08096       REINITIALIZE BUFFER ADDR IN REG 9
05,1,15,14             LOAD AND INDEX BY 1
15,1,15,11             STORE RA INTO Y* AND INDEX Y*
02,0,13,11             DECREMENT COUNTER
45,1,374               COUNTER EQUAL ZERO
22,2,11,00,00026       ADD 26 TO BUFFER ADDR
01,2,13,00,00040       RESET  BUFFER WORD COUNT
02,0,15,11             DECREMENT NEXT COUNTER (NO. OF BUFFERS)
45,1,356               COUNTER EQUAL ZERO
63,0,15,00             LOAD REG 13 WITH 0
03,0,15,05             LOAD SR1 WITH 0
01,3,01,00,02030    ***CRAMER'S RULE ALGORITHM***  STORE X OF EQN 1
01,3,03,00,02031       STORE Y COEFF, EQN 1, INTO REG 03
01,3,05,00,02032       STORE CONSTANT, EQN 1, INTO REG 05
01,3,11,00,02033       STORE Y COEFF, EQN 2, INTO REG 11
01,3,13,00,02034       STORE X COEFF, EQN 2, INTO REG 13
01,3,15,00,02035       STORE CONSTANT, EQN 2, INTO REG 15
26,0,00,13             MULT LEFT DIAG OF DENOM, RESULT IN 00/01
26,0,10,03             MULT RT.  DIAG OF DENOM, RESULT IN 10/11
21,0,00,10             SUBTRACT DOUBLE, RESULT OF DENOM IN 00/01
25,3,00,17,02048       COMPARE DENOM WITH ZERO
44,1,052               JUMP, IF EQUAL (ZERO DETERMINANT)
12,3,00,06,00014       STORE DOUBLE, DENOM INTO 16/17
01,2,00,00,00030       LOAD 00 INTO REG 00
01,3,01,00,02030       RESTORE Y COEFF, EQN 1, INTO REG 01
01,3,03,00,02031       RESTORE X COEFF, EQN 1, INTO REG 03
01,3,05,00,02032       RESTORE CONSTANT OF EQN 1 IN REG 05
01,3,11,00,02033       RESTORE Y COEFF OF EQN 2 IN REG 11
01,3,13,00,02034       RESTORE X COEFF OF EQN 2 IN REG 13
01,3,15,00,02035       RESTORE CONSTANT OF EQN 2 IN REG 15
26,0,00,15             X NUM LEFT DIAG MULT, RESULT IN 00/01
26,0,10,05             X NUM RT.  DIAG MULT, RESULT IN 10/11
21,0,00,10             X NUM RESULT IN REG 00/01
27,0,00,17             X RESIDES IN REG 01
26,0,04,13             Y NUM LEFT DIAG MULT, RESULT REG 04/05
26,0,14,03             Y NUM RT.  DIAG MULT, RESULT REG 14/15
21,0,04,14             DOUBLE SUBTRACT (EVAL NUM), RESULT 04/05
27,0,04,17             DIVIDE NUM/DENOM, Y RESIDES IN 05
```

```
01,2,07,00,16339        ***INT TO BCL CONVERSION*** TWICE BUFFER ADDR PLUS 1
63,0,03,00                 LIT LOAD, C INTO REG 0
63,3,00,12                 LIT DIVIDE, REG 0/1 BY 10
10,3,03,07,00030           BYTE STORE, REG 7 MUST CONTAIN TWICE ADDR
44,1,033                   LOCAL JUMP EQUAL (ZERO QUOTIENT)
63,0,03,00                 LIT LOAD, C INTO REG 0
62,0,07,01                 LIT SUBTRACT, 1 FROM REG 7
40,1,371                   LOCAL JUMP
01,2,07,00,16603           LOAD CONSTANT, TWICE ADDR PLUS 1 IN REG. 7
63,0,04,00                 LIT LOAD, C INTO REG 4
63,3,04,12                 LIT DIVIDE, REG 4/5 BY 10
10,3,04,07,00030           BYTE JUMP
44,1,033                   JUMP EQUAL, CHECK FOR ZERO QUOTIENT
63,0,04,00                 LIT LOAD, 0 INTO REG 4
62,0,07,01                 LIT SUBTRACT, 1 FROM REG 7
40,1,371                   LOCAL JUMP
63,0,15,01                 LOAD REG 13 WITH A 1
03,0,15,05                 LOAD SR1 WITH A 1
12,1,04,17                 STORE DOUBLE INTO IOCW
35,0,03,00                 DUMP 5 BUFFERS TO PRT
03,0,05,00                 HALT
63,0,15,01              ***NO SOLUTION CASE***
03,0,15,05                 LOAD SR1 WITH A 1
20,2,11,00,00178           RECOMPUTE OUTPUT BUFFER ADDR FOR 'NO SOLUTION'
01,2,13,00,00040           REINITIALIZE COUNTER
05,1,15,14                 LOAD AND INDEX BY 1
15,1,15,11                 STORE RA INTO Y* AND INDEX Y*
02,0,13,11                 DECREMENT COUNTER
45,1,374                   COUNTER EQUAL ZERO
12,1,04,17                 STORE DOUBLE INTO IOCW
35,0,03,00                 DUMP 5 BUFFERS TO PRT
03,0,05,00                 HALT
```

```
        G                                   COMMENCE EXECUTION


THIS IS A DEMONSTRATION PROGRAM OF THE AN/UYK20 EMULATOR.
THE PROGRAM SOLVES TWO SIMULTANEOUS EQUATIONS BY CRAMER'S RULE.


THE INPUT COEFFICIENT VALUES AND CONSTANTS FOR EACH EQUATION MUST BE INSERTED
USING LOADER CONTROL CARDS IN THE FOLLOWING LOCATIONS

1ST EQUATION.   A1 IN LOCATION 2050 D.
                B1 IN LOCATION 2051 D.
                CONSTANT IN LOCATION 2052 D.

2ND EQUATION.   A2 IN LOCATION 2053 D.
                B2 IN LOCATION 2054 D.
                CONSTANT IN LOCATION 2055 D.


     THE RESULT IS.
     X >        2

     Y >        1
TO CHANGE VARIABLES INSERT LOADER JCL HERE, OTHERWISE EXIT
        I2    L1        00090            REINITIALIZE BUFFER WORD COUNT FOR BUFFER INIT.
        D  02053        00031            Y COEFF OF 2ND EQUATION
        D  02054        00031            X COEFF OF 2ND EQUATION
        D  02055        00033            CONSTANT OF 2ND EQUATION
        G  00031                         RE-EXECUTE PROGRAM AT ADDR 1055 DECIMAL
     THE RESULT IS.
     X >
        NO SOLUTION
     Y >
TO CHANGE VARIABLES INSERT LOADER JCL HERE, OTHERWISE EXIT
        E                                END EXECUTION
```

APPENDIX F. EMULATOR LISTING

This appendix provides a listing of the TRANSLANG assembler output of the AN/UYK-20 emulation. A source file copy of the emulator exists on disk as well as on cards. A microinstruction object file is also maintained on disk.

The listing is divided into four sections. The left most section contains the microinstruction address composed of four hexadecimal digits followed by a 56-bit microinstruction created from a TRANSLANG instruction. The center section contains the TRANSLANG source which includes labels, an instruction, and/or a comment field. The final number printed on the right most side of the listing is the sequence number of the TRANSLANG source statement. This number, printed in decimal, is created by a Burroughs software utility program called CARD-LIST. This number must be used when editing source programs on disk.

```
                                        $MERGE TED/UYK20
                                        PROGRAM TED/UYK20-OBJECT                                                    0C001C00 D
                                        % * * * * * * * * * * * * MAIN MEMORY ADDRESS MAPPING * * * * * * * * * * 00002C00 D
                                                                                                                    00003C00 D
                                        GREG1 = 0                       % ADDRESS OF GENERAL REGISTER STACK ONE     00004000 D
                                        GREG2 = 16                      % ADDRESS OF GENERAL REGISTER STACK TWO     00005000 D
                                        PSW = 32                        % PROGRAM STATUS WORD (PAR & SR #1)         00006C00 D
                                        BKPT = 33                       % BREAKPOINT REGISTER                       00007C00 D
                                        STATUS2 = 34                    % STATUS REGISTER TWO                       00008C00 D
                                        NXTINSTR = 35                   % ADDRESS OF NEXT INSTRUCTION               00009000 D
                                        CLOCKTIME = 36                  % TIME OF EXECUTION                         00010C00 D
                                        RTC = 37                        % REAL TIME CLOCK                           00011C00 D
                                        WORKSPACE = 38                  % TEMPORARY STORAGE STACK                   00012C00 D
                                        STACK = 44                      % TEMPORARY STORAGE STACK                   00013C00 D
                                        IOCW = 50                       % INPUT OUTPUT COMMAND WORDS (THRU 53)      00014C00 D
                                        HEXADDR = 120                   % HEX ADDR TO BE PRINTED (TWO WORD VAL)     00015C00 D
                                                                        %                                          00016C00 D
                                                                        % CARDBUFFER ADDRESSES                      00017C00 D
                                                                        %                                          00018C00 D
                                        CARDBUF = 122                   % ADDRESS OF CARD BUFFER                    00019C00 D
                                        CARD1X4 = 122                   % ADDRESS OF CARD COLS 1-4                   00020C00 D
                                        CARD5X8 = 123                   % ADDRESS OF CARD COLS 5-8                   00021C00 D
                                        CARD9X12 = 124                  % ADDRESS OF CARD COLS 9-12                  00022C00 D
                                        CARD13X16 = 125                 % ADDRESS OF CARD COLS 13-16                 00023C00 D
                                        CARD77X80 = 141                 % ADDRESS OF CARD COLS 77-80. END OF CARD00024C00 D
                                                                        %                                          00025C00 D
                                        PRINTBUFF = 153                 % PRINT BUFFER LOCATION                     00026C00 D
                                                                        %                                          00027C00 D
                                        CRTBUFF = 186                   % CRT BUFFER (20 WORDS)                     00028C00 D
                                                                        %                                          00029C00 D
                                        ERRORLIST = 206                 % LIST OF POSSIBLE ERRORS                   00030C00 D
                                        PAGEREG = 768                   % PAGE ADDRESS REGISTERS STACK BASE         00031C00 D
                                                                        % (0-255)                                  00032C00 D
                                                                        %                                          00033C00 D
                                        % * * * * * * * * * * * * * * LOADER * * * * * * * * * * * * * * * * * * * *00034000 D
                                                                        %                                          00035C00 D
                                                                        % BEGIN PROGRAM BY ZEROING MAIN MEMORY      00036C00 D
                                        START:                          % RESTART RETURN POINT                      00037C00 D
                                                                        % IF CRT INPUT DESIRED, PUT IRQ SW ON       00038C00 D
0C00  4800 C000 0030 C0F0                       WAIT                    % TOGGLE EXECUTION BY SINGLE STEP           00039C00 D
0C01  0088 00C1 401C 00F0                       SET GC1; WHEN GC1 THEN 1 L = A1,MAR2                               00040C00 D
0C02  0000 0000 0030 C020                       COMP 16 = SAR                                                      00041000 D
0C03  4812 0000 0030 00F0                       0 = MIR; SAVE                                                      00042C00 D
0C04  9809 ACD= 4000 1CF0                       MW2; A1 -1 = A1; IF SAI                                            00043C00 D
0C05  6819 0000 0030 C0FC                       IF ABT THEN SKIP ELSE STEP  % WHEN A1 = ZERO 64K OF MEMORY = 0 0C044C00 D
0C06  9C28 A0C0 001C 00FC                       WHEN SAI THEN A1 = MAR2; JUMP                                      00045C00 D
0C07  0000 0000 0030 C040                       NEWCARD - 1 = MPCR      % FETCH A CARD                             00046C00 D
                                                                        %                                          00047C00 D
                                                                        %                                          00048C00 D
                                                                        %                                          00049C00 D
                                        CONTROLCD:                      % THIS ROUTINE WILL READ COLUMNS 1-4 AND 00050C00 D
                                                                        % EXAMINE COLUMN 1 FOR A CONTROL CARD       00051C00 D
                                                                        % IDENTIFIER.  IF THE CARD IS A CONTROL     00052C00 D
                                                                        % THE APPROPRIATE ROUTINE WILL BE CALLED.00053C00 D
0C08  4809 20C0 001C C0F0                       LIT = MAR2              % LOAD ADDRESS OF CARD COLUMNS 1-4.         00054C00 D
0C09  07AC 00C0 0030 C0EC                       CARD1X4 = LIT                                                      00055C00 D
0C0A  0010 00C0 0030 0060                       INPUT - 1 = CPCR        % READ COLUMNS 1-4                         00056C00 D
0C0B  4809 20C1 C030 40F0                       LIT C = A1,CSAR         % LOAD MSB OF A1 WITH 00111111              00057C00 D
```

```
0C0C  03EC 0000 0000 0090   63 = LIT B = SAR
0C0D  4809 AC56 8B3C 00F0   A1 AND B R = B          % FILL LSB OF P REGISTER WITH CHARACTER   00058C00 D
                                                    % FROM COLUMN ONE.                        00060C00 D
                                                    %                                         00061C00 D
0C0E  4809 2C52 0030 00F0   LIT EQL B                                                         00062C00 D
0C0F  0120 0000 0030 00E0   18 = LIT                % BCL CODE FOR "B"                         00063C00 D
0C10  680B 2C52 0030 C0F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "B"           00064000 D
0C11  002C 0000 0030 0040   BREAKPOINT - 1 = MPCR   % JUMP TO BREAKPOINT ROUTINE              00065C00 D
                                                    %                                         00066C00 D
0C12  0140 C000 0030 00E0   20 = LIT                % BCL CODE FOR "D"                         00067C00 D
0C13  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "D"           00068C00 D
0C14  003C 0000 0030 0040   DATA - 1 = MPCR         % JUMP TO DATA ROUTINE                    00069000 D
                                                    %                                         00070C00 D
0C15  0150 0000 0030 00E0   21 = LIT                % BCL CODE FOR "E"                         00071C00 D
0C16  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "E"           00072C00 D
0C17  FFFC 0000 0030 0050   START - 1 = MPCR        % JOB TERMINATION. REINITIALIZE MEMORY    00073000 D
                                                    %                                         00074C00 D
0C18  0170 0000 0030 00E0   23 = LIT                % BCL CODE FOR "G"                         00075C00 D
0C19  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "G"           00076C00 D
0C1A  004C 0000 0030 0040   GO - 1 = MPCR           % JUMP TO GO ROUTINE                      00077C00 D
                                                    %                                         00078C00 D
0C1B  0190 0000 0030 00E0   25 = LIT                % BCL CODE FOR "I"                         00079C00 D
0C1C  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "I"           00080C00 D
0C1D  0050 0000 0000 0040   SETINDEXREG - 1 = MPCR  % JUMP TO THE SET INDEX REG ROUTINE       00081C00 D
                                                    %                                         00082C00 D
0C1E  023C 0000 0000 00E0   35 = LIT                % BCL CODE FOR "L"                         00083C00 D
0C1F  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "L"           00084C00 D
0C20  006C 0000 0030 0040   LOAD - 1 = MPCR         % JUMP TO LOAD ROUTINE                    00085C00 D
                                                    %                                         00086C00 D
                                                    %                                         00087C00 D
0C21  0270 0000 0030 00E0   39 = LIT                % BCL CODE FOR "P"                         00088C00 D
0C22  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "P"           00089C00 D
0C23  0070 0000 0030 0040   SETPAGING - 1 = MPCR    % JUMP TO SET PAGING BIT ROUTINE          00090C00 D
                                                    %                                         00091C00 D
0C24  0290 0000 0000 00E0   41 = LIT                % BCL CODE FOR "R"                         00092C00 D
0C25  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "R"           00093C00 D
0C26  0080 0000 0000 0040   RESERVESPACE - 1 = MPCR  % JUMP TO RESERVESPACE ROUTINE           00094000 D
                                                    %                                         00095C00 D
0C27  032C 0000 0030 00E0   50 = LIT                % BCL CODE FOR "S"                         00096C00 D
0C28  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "S"           00097C00 D
0C29  0090 0000 0000 0040   SWITCHES - 1 = MPCR     % JUMP TO SWITCHES ROUTINE                00098C00 D
                                                    %                                         00099C00 D
0C2A  033C 0000 0000 00E0   51 = LIT                % BCL CODE FOR "T"                         00100C00 D
0C2B  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "T"           00101C00 D
0C2C  00A0 0000 0030 0040   TRACE - 1 = MPCR        % JUMP TO TRACE ROUTINE                   00102C00 D
                                                    %                                         00103C00 D
0C2D  0130 0000 0030 00E0   19 = LIT                % BCL CODE FOR "C"                         00104C00 D
0C2E  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "C"           00105C00 D
0C2F  00B0 0000 0030 0040   CHARSTRING - 1 = MPCR   % JUMP TO CHARACTER STRING ROUTINE        00106C00 D
                                                    %                                         00107C00 D
0C30  0300 0000 0030 00E0   45 = LIT                % BCL CODE FOR BLANK                       00108C00 D
0C31  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR BLANK         00109C00 D
0C32  00C0 0000 0030 0040   INSTRUCTION - 1 = MPCR  % JUMP TO THE INSTRUCTION HANDLING SUBR00110C00 D
                                                    %                                         00111C00 D
0C33  0240 0000 0030 00E0   36 = LIT                % BCL CODE FOR "M"                         00112C00 D
0C34  680B 2C52 0030 00F0   LIT EQL B; IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "M"           00113C00 D
0C35  00D0 0000 0030 0040   MACHSTAT - 1 = MPCR     % JUMP TO MACHINE STATUS ROUTINE          00114C00 D
                                                    %                                         00115C00 D
0C36  00EC 0000 0030 0040   NEWCARD - 1 = MPCR      % DEFAULT. GET NEW CARD                   00116C00 D
                                                    %                                         00117000 D
```

```
                                LOAD:                   % SET UP MEMORY INTO 256 PAGES OF 256     00118C00 D
                                                        % WORDS (32 BIT) EACH.  THE PAGE NO. IN   00119C00 D
                                                        % THE "L" CARD (COL. 5-8) WILL BE         00120C00 D
                                                        % TRANSLATED INTO  THE BASE ADDRESS OF    00121C00 D
                                                        % THE SELECTED PAGE AND STORED IN         00122C00 D
                                                        % THE HEXTINST REG TO BE UTILIZED AS THE  00123C00 D
                                                        % BASE ADDRESS TO LOAD THE PROGRAM.       00124C00 D
0C37  4809 0000 0B30 00FC       0 = B,MIR               % ZERO B AND MIR                          00125C00 D
0C38  4809 0643 001C C0F0       AMPCR = MAR2            %SET UP ADDRESS OF PAGE REGISTER          00126C00 D
0C39  3000 0000 0030 00C0       PAGEREG = AMPCR                                                   00127C00 D
0C3A  0FE0 0000 0030 00E0       254 = LIT               % SET UP CTR FOR 255 + 1 ITERATIONS       00128C00 D
0C3B  4812 0000 0C01 00F0       LCTR; SAVE              % THIS LOOP WILL INITIALIZE 256 PAGE ADDR00129C00 D
                                                        % REGISTERS TO THEIR BASE ADDRESSES.      00130C00 D
0C3C  9809 C000 0030 1CF0       MW2; BMI; IF SAI        % TRANSFER MIR TO B AFTER MEMORY WRITE    00131C00 D
0C3D  9C08 0C46 0032 00F0       WHEN SAI THEN B + 1 = MIR,INC % INCREMENT PAGE NO. TO BE LOADED00132C00 D
                                                        % INTO THE NEXT PAGE ADDRESS              00133C00 D
                                                        % AND INCREMENT LOOP COUNTER              00134C00 D
0C3E  8429 0F46 001C 00FC       IF NOT COV THEN BMAR + 1 = MAR2; JUMP  % INCREMENT ADDR OF PAGE00135C00 D
0C3F  4809 2000 001C 00F0       LIT = MAR2              % LOAD ADDR OF COL 1-4                    00136C00 D
0C40  07A0 0000 0030 00EC       CARD1X4 = LIT                                                     00137C00 D
0C41  00F0 0000 003C C060       INPUT - 1 = CPCR        % READ COLS 1-4                           00138C00 D
0C42  0100 0000 0030 0060       GETADDR - 1 = CPCR      % GET OCTAL VALUE OF PAGE NO. IN COL.6-8 00139C00 D
0C43  4809 0C41 0030 00F0       B L = MIR               % MULT PAGE NO. BY 256 TO GET BASE ADDR   00140C00 D
0C44  0230 0000 0000 0060       NXTINSTR = LIT; COMP B = SAR  % INTO NXTINSTR REG WHICH WILL      00141C00 D
0C45  4809 2000 001C 00F0       LIT = MAR2              % WRITE THE BASE ADDRESS OF SELECTED PAGE00142C00 D
0C46  0110 0000 0030 004C       OUT - 1 = MPCR          % TO BE USED AS THE BASE ADDRESS TO LOAD 00143C00 D
                                                        % THE PROGRAM.                            00144C00 D
                                                        %                                         00145C00 D
                                                        %                                         00146C00 D
                                INPUT:                  % THIS ROUTINE WILL READ THE CONTENT OF   00147C00 D
                                                        % THE MEMORY ADDRESS IN MAR2.             00148C00 D
0C47  A809 0000 003C 0CF0       MR2; IF RDC             % PERFORM READ                            00149C00 D
0C48  AC28 0000 0C00 00F0       WHEN RDC THEN BEX; JUMP  % READ CONTENTS OF MAR2 INTO B.          00150C00 D
                                                        %                                         00151C00 D
                                                        %                                         00152C00 D
                                GO:                     % THIS ROUTINE RETRIEVES THE PAR CONTENTS00153C00 D
                                                        % AND STATUS REG 1 CONTENTS AND PACKS     00154C00 D
                                                        % THEM INTO A1 AND THEN JUMPS TO THE      00155C00 D
                                                        % EMULATOR                                00156C00 D
0C49  0120 0000 0030 0060       GETADDR - 1 = CPCR      % GET ADDRESS IN COL. 5-8 OF "GO" CARD    00157C00 D
0C4A  4809 00C1 4C0C 00F0       1 L = A1                                                          00158C00 D
0C4B  A200 C000 0C30 C0A0       COMP 10 = SAR; PSW = LIT                                          00159C00 D
0C4C  4809 AC40 4030 00F0       A1 + B = A1             % OFFSET ADDRESS BY 1024                  00160C00 D
0C4D  4809 2000 001C C0F0       LIT = MAR2              % LOAD ADDRESS OF PSW                     00161C00 D
0C4E  0460 C000 0000 0060       INPUT - 1 = CPCR        % RETRIEVE CONTENTS OF PSW                00162C00 D
0C4F  4809 AC50 4030 00F0       A1 OR B = MIR, A1       % STORE COMPLETE PSW INTO MIR AND A1      00163C00 D
0C50  0130 0000 0030 0060       OUTPUT1 - 1 = CPCR      % LOAD PAR REG WITH START ADDRESS OF PROG00164C00 D
0C51  014C 0000 0000 004C       STARTIME - 1 = MPCR     % START EMULATION OF UYK20                00165C00 D
                                                        % START CLOCK AND FETCH FIRST INST.       00166C00 D
                                                        %                                         00167C00 D
                                                        %                                         00168C00 D
                                                        %                                         00169C00 D
                                GETADDR:                %GET ADDRESS FIELD FROM COLUMNS N-M       00170C00 D
                                                        %OF INPUT CARD. ADDRESS WILL BE IN        00171C00 D
                                                        %DECIMAL. (N & M STAND FOR CARD COLS)     00172C00 D
                                                        %                                         00173C00 D
0C52  4809 0000 0030 00F0       BEX                     % GET PRECEDING MEMORY WORD FROM SWI      00174C00 D
0C53  4809 0640 2000 00F0       AMPCR = A2              %SAVE RETURN                              00175C00 D
0C54  4809 0C41 0B30 40FC       B L = B, CSAR                                                     00176C00 D
0C55  0300 0000 0030 0090       COMP 24 = SAR;48=LIT    %ISOLATE LSB OF CARD(N-3)XN               00177C00 D
```

```
0056  9809 0040 0030 00F0          B R = A1                                  00178000 D
0057  9809 A152 0030 00FC          A1 EQL LIT          % TEST FOR BLANK ADDR FIELD   00179000 D
0058  5C0B 0000 0B30 00F0          IF TRUE THEN 0 = B; STEP ELSE SKIP  % IF BLANK ADDR TREAT AS 0 00180000 D
0059  0150 0000 0030 0040          ALLBLANKS - 1 = MPCR  %OUTPUT EQUAL 0 IN B REG   00181000 D
005A  0160 0000 0030 C06C          PACKED-1 = CPCR     %PACK DECIMAL NUMBERS INTO SINGLE WORD 00182000 D
005B  0170 0000 0000 0060          DECOCT-1 = CPCR     %DECODE  THE DECIMAL NUMBERS INTO OCTAL 00183000 D
                                ALLBLANKS:                                   00184000 D
005C  9809 C000 0030 00FC          A2 = AMPCR                                00185000 D
005D  9809 0000 0030 00F0          STEP                % NEEDED FOR AMPCR ASSIGN PRIOR TO JUMP 00186000 D
005E  482D 0000 0030 00F0          JUMP                                      00187000 D
                                                       %                     00188000 D
                                OUT:                   % THIS ROUTINE PERFORMS A MEMORY WRITE 00189000 D
005F  9809 0000 0030 1CF0          MW2; IF SAI                               00190000 D
0060  9C08 0000 0030 00F0          WHEN SAI THEN 0; STEP                     00191000 D
                                                       %                     00192000 D
                                                       %                     00193000 D
                                NEWCARD:               %CALL IOP TO READ NEXT CARD THEN 00194000 D
                                                       %JUMP TO CONTROL SPACE CARD ROUTINE. 00195000 D
                                                       %THIS ROUTINE INSERTS SPACES INTO 00196000 D
                                                       %HEXADDR AND CARDBUFFER          00197000 D
                                                       %REMAINING 11 WORDS ARE FILLED WITH 00198000 D
                                                       %SPACES FOR ECHO PRINT ROUTINE.  00199000 D
0061  9809 00C0 0931 00F0          0 = BCB,LCTR                              00200000 D
0062  01F0 0000 0030 00E0          31 = LIT            %33 WORDS WILL HAVE SPACES 00201000 D
0063  9809 20C0 001C 00F0          LIT = MAR2                                00202000 D
0064  0780 0000 0030 00A0          HEXADDR = LIT; COMP 16 =SAR               00203000 D
0065  4809 0040 0030 00F0          B = MIR             %PUT SPACES INTO MIR  00204000 D
                                                       %                     00205000 D
                                                       %                     00206000 D
0066  0180 00C0 0000 0060       SPACES:  OUTPUT1 - 1 = CPCR  %WRITE SPACES INTO MEMORY 00207000 D
0067  840B 0F45 001E C0F0          IF NOT COV THEN BMAR + 1 = MAR2,INC; STEP ELSE SKIP 00208000 D
0068  065C 0000 0030 0040          SPACES-1 = MPCR                           00209000 D
                                                       %                     00210000 D
                                                       %                     00211000 D
                                CARDFETCH:             % RETRIEVE CARD        00212000 D
0069  9809 20C0 0B30 00F0          LIT = B; IF SAI                           00213000 D
006A  07AC 0000 0000 00E0          CARDBUF = LIT       % CARDBUFF ADDRESS INTO B 00214000 D
006B  980E 0000 0030 00FC          IF IRQ THEN STEP ELSE SKIP  % IRQ INDICATES CRT INPUT 00215000 D
006C  019C 00C0 0030 0060          CRTIN - 1 = CPCR                          00216000 D
006D  4809 0040 0030 00F0          B = MIR             % CONTENTS FOR MAILBOX 00217000 D
006E  01AC 00C0 0030 0060       BCRD:    EXTOP - 1 = CPCR                    00218000 D
006F  0600 00C0 0030 0040          BCRD - 1 = MPCR                           00219000 D
0070  980B 00C0 0030 00F0          IF IRQ THEN STEP ELSE SKIP  % IRQ -> FALSE THEN ECHO PRT (PTR) 00220000 D
0071  007C 0000 0000 0040          CONTROLCD - 1 = MPCR  % JUMP TO CONTROL CARD ROUTINE 00221000 D
0072  9809 20C0 0030 00F0          LIT = MIR                                 00222000 D
0073  078C 00C0 0000 00EC          HEXADDR = LIT                             00223000 D
0074  9809 00C1 0F30 00F0          1 L = BBI                                 00224000 D
0075  0600 00C0 0030 0020          COMP 16 = SAR                             00225000 D
0076  4809 0040 0030 00F0          B = MIR             %MIR CONTAINS 1/HEXADDR 00226000 D
                                                       %                     00227000 D
                                                       %                     00228000 D
                                ECHOPRINTCD:           % ECHO CARD READ      00229000 D
0077  0180 00C0 0000 0060          EXTOP - 1 = CPCR    % JUMP TO EXTOP ROUTINE 00230000 D
0078  0760 00C0 0030 0C40          ECHOPRINTCD - 1 = MPCR % UNSUCCESSFUL IO OPERATION 00231000 D
0079  007C 0000 0000 0040          CONTROLCD - 1 = MPCR  % JUMP TO CONTROL CARD ROUTINE 00232000 D
                                                       %                     00233000 D
                                                       %                     00234000 D
                                EXTOP:                 %THIS ROUTINE PERFORMS THE IOP INTERFACE 00235000 D
                                                       %BY PASSING TO THE IOP THE FUNCTION 00236000 D
                                                       %AND THE ADDRESS IN THE MAILBOX. THE IOP 00237000 D
```

```
                                                                  %IN THE B REGISTER.                                  00239000 D
0C7A  1C88 C002 001C 00F0                SET GC2 WHEN GC2 THEN NOT 0 = MAR2                                           00240000 D
0C7B  9809 00C0 0030 1CF0                MW2; IF SAI                                                                  00241000 D
0C7C  9F08 0000 0030 00F0                WHEN SAI THEN 0; SET INT                                                     00242000 D
0C7D  0809 0000 0C30 C0F0                IF INT                                                                       00243000 D
0C7E  0808 00C0 0030 00FC                WHEN INT THEN STEP                                                           00244000 D
0C7F  A809 00C0 0C30 0CF0                BEX; MR2; IF ROC                                                             00245000 D
0C80  AEC8 0C42 000C C0FC                WHEN ROC THEN NOT B; RESET GC2                                               00246000 D
0C81  682F 0000 0030 00F0                IF ABT THEN JUMP ELSE RETN                                                   00247000 D
                                                                  %                                                    00248000 D
                                                                  %                                                    00249000 D
                                OUTPUT1:                                                                               00250000 D
0C82  9809 0000 0C30 1CF0                MW2; IF SAI                                                                  00251000 D
0C83  9C28 00C0 0030 00F0                WHEN SAI THEN 0; JUMP                                                        00252000 D
                                                                  %                                                    00253000 D
                                                                  %                                                    00254000 D
                                                                  %                                                    00255000 D
0C84  4809 2001 200C 00F0       CRTIN:   LIT L = A2                % CRT READ CODE FOR CARDFETCH ROUTINE                00256000 D
0C85  0060 0000 0030 00AC                COMP 16 = SAR; 6 = LIT                                                       00257000 D
0C86  4809 CC50 0B30 C0F0                A2 OR B = B                                                                  00258000 D
0C87  482D 0000 0C00 00FC                JUMP                                                                         00259000 D
                                                                  %                                                    00260000 D
                                                                  %                                                    00261000 D
                                PACKED:                           % THIS ROUTINE MUST BE PASSED A 32 BIT                00262000 D
                                                                  % WORD IN UNPACKED DECIMAL FORM(4 DIGIT              00263000 D
                                                                  % DECIMAL NUMBER).  THE NUMBER IS PASSED             00264000 D
                                                                  % IN A1.  THE PACKED DECIMAL EQUIVALENT              00265000 D
                                                                  % OF THE NUMBER IS RETURNED IN B REGISTER0C266000 D
                                                                  % WHICH CONSISTS OF AN 8 DIGIT DECIMAL NOCC267000 D
0C88  49C9 0000 0C30 00F0                SET LC1                  % THIS STEP IS OMITTED FOR A 4 DIGIT NC. 00268000 D
0C89  4809 0000 0C30 00F0                0 = MIR                  % ZERO MIR                                           0C269000 D
008A  4809 00C0 0001 C0F0       NEXT4:   LCTR                     % SET COUNTER TO PERFORM LOOP 4 TIMES                00270000 D
0C8B  3030 0000 0030 C080                3 = LIT; COMP 4 = SAR                                                        0C271000 D
                                PACKTOP:                                                                               00272000 D
0C8C  4809 A001 4C30 00F0                A1 L = A1                % ELIMINATE HIGH ORDER 4 BITS OF MSB CHAR0C273000 D
0C8D  4809 A000 8F30 00F0                A1 R = BBI               % SHIFT A1 RIGHT 28 BITS AND "OR" WITH               00274000 D
                                                                  % MIR AND PLACE IN B REGISTER                        00275000 D
0C8E  4809 0C41 0032 00F0                B L = MIR,INC            % LEFTMOST DIGIT NOW IN HIGH ORDER                    00276000 D
                                                                  % POSITION OF MIR.                                   00277000 D
0C8F  940E A0C1 4030 00F0                IF NOT COV THEN A1 L = A1; STEP ELSE SKIP  % SHIFT NEXT DIGIT                00278000 D
                                                                  % INTO POS. FOR NEXT PASS THROUGH LOOP               00279000 D
0C90  08BC 00C0 0030 C040                PACKTOP - 1 = MPCR       %BRANCH TO TOP OF LOOP                               00280000 D
0C91  2819 0000 0000 00FC                IF LC1 THEN SKIP         % CHECK TO SEE IF AN 8 DIGIT NUMBER                  00281000 D
0C92  01C0 0000 0030 004C                PACKEND - 1 = MPCR       % JUMP TO THE END OF THE ROUTINE                     0C282000 D
                                                                  % IF ONLY A 4 DIGIT NUMBER WAS INPUT.               0C283000 D
0C93  4809 0F45 001C 00F0                BMAR + 1 = MAR2          %PREPARE MAR TO READ NEXT 4 DIGITS                   00284000 D
0C94  A809 00C0 0C30 0CFC                BEX; MR2; IF ROC         % B NOW CONTAINS LOW ORDER 4 DIGITS                  00285000 D
                                                                  % OF AN 8 DIGIT NUMBER.                              C0286000 D
0C95  AC08 0C40 4030 00F0                WHEN ROC THEN B = A1     % WHEN READ COMPLETE, SET A1 EQUAL TO B              0C287000 D
0C96  0890 00C0 0030 0040                NEXT4 - 1 = MPCR         % BRANCH TO THE TOP OF THE LOOP                      00288000 D
                                PACKEND:                                                                               00289000 D
0C97  482D 0000 0C30 C0F0                JUMP                     %RETURN TO CALLING ROUTINE WITH PACKED               0C290000 D
                                                                  %NUMBER IN B REGISTER                                00291000 D
                                                                  %                                                    0C292000 D
                                                                  %                                                    0C293000 D
                                DECOCT:                           % THIS ROUTINE WILL CONVERT DECIMAL                   0C294000 D
                                                                  % NUMBERS INTO THEIR OCTAL EQUIVALENTS.              0C295000 D
                                                                  % THIS ROUTINE REQUIRES A PACKED DECIMAL             0C296000 D
                                                                  % NUMBER PASSED VIA THE B REGISTER.  THE             0C297000 D
```

```
                                                        % OCTAL EQUIVALENT WILL BE RETURNED IN B.0C298C00 D
0C98  4809 0C40 4C30 00F0          B = A1               % PACKED RESULTS ARRIVE IN B REG          0C299C00 D
0C99  4809 00C0 0800 00F0          0 = B                % CLEAR B REGISTER                        0C300C00 D
0C9A  4809 00C0 0031 00F0          C = MIR, LCTR        % SET COUNTER FOR 7 ITERATIONS.           00301C00 D
0C9B  0060 00C0 0000 00E0          6 = LIT                                                        C0302C00 D
                                                        % PLACE DECIMAL NUMBER IN A1 AND CLEAR B.00303C00 D
0C9C  4809 A000 9030 00F0  TOPDEC: A1 R = A3            % PUT HIGH ORDER 4 BITS OF A1 INTO        C0304C00 D
0C9D  800C 00C0 0030 0C30          COMP 4 = SAR         % LOW ORDER 4 BITS OF A3.                00305C00 D
0C9E  4809 A0C1 4C30 00F0          A1 L = A1            % PREPOSITION NEXT DECIMAL DIGIT IN A1.   0C306C00 D
0C9F  4809 EC41 1830 00F0          A3 + B L = A3, BAD   % MULT DECIMAL DIGIT BY 8               0C307C00 D
0CA0  30CC 0000 0030 0030          COMP 3 = SAR                                                   0C308C00 D
0CA1  4809 CC41 0832 00F0          B L = B,INC          % MULT DECIMAL DIGIT BY 2, INCREMENT CTR 00309C00 D
00A2  3000 00C0 0030 C030          COMP 1 = SAR         %EFFECTIVELY MULT BY A FACTOR OF 10.     0C310C00 D
0CA3  4809 EC43 0B30 00F0          A3 + B = B           % PARTIAL SUM OF CONVERSION               0C311C00 D
0CA4  9C19 A000 9030 00F0          IF COV THEN A1 R = A3, SKIP  % GET LS DIGIT FROM A1 IF DONE  0C312C00 D
0CA5  0980 00C0 0030 0040          TOPDEC - 1 = MPCR    % GO TO TOP OF LOOP                      00313C00 D
0CA6  300C 0000 0030 0030          23 = SAR                                                       0C314C00 D
0CA7  4809 EC40 0B30 00F0          A3 + B = B           % ADD LS DIGIT TO B AND RETURN OCTAL      0C315C00 D
                                                        % VALUE IN B REGISTER.                   00316C00 D
0CA8  482D 00C0 0030 00F0          JUMP                                                           00317C00 D
                                                        %                                        C0318C00 D
                                                        %                                        00319C00 D
                           DATA:                        % CONVERT DECIMAL DATA IN COL 13-20 INTO 0032CC00 D
                                                        % OCTAL AND STORE AT THE ADDRESS IN       00321C00 D
                                                        % COL 4-8                                 0C322C00 D
0CA9  051C 0000 0030 0060          GETADDR - 1 = CPCR   % RETRIEVE ADDR IN COL 4-8 IN OCTAL      00323C00 D
0CAA  4809 0C40 1030 00F0          B = A3               % TRANSFER ADDR TO A3 REG                 00324C00 D
0CAB  4809 2CC0 001C 00F0          LIT = MAR2           % LOAD ADDRESS OF COL 13-16               0C325C00 D
0CAC  0700 00C0 0030 C0E0          CARD13X16 = LIT                                                0C326C00 D
0CAD  046C 00C0 0030 0060          INPUT - 1 = CPCR     % READ IN COL 13-16                      0C327C00 D
0CAE  4809 0C40 4C30 00F0          B = A1                                                         0C328C00 D
0CAF  0870 00C0 0000 0060          PACKED - 1 = CPCR    % PACK DATA IN COL 13-20                 0C329C00 D
0CB0  4809 E0C0 2030 00F0          A3 = A2              % SWITCH TEMP ADDRESS STORAGE             0C330C00 D
0CB1  097C 00C0 0000 0060          DECOCT - 1 = CPCR    % CONVERT DECIMAL TO OCTAL               0C331C00 D
0CB2  4809 0C40 0030 00F0          B = MIR              % MIR CONTAINS DATA IN OCTAL              0C332C00 D
0CB3  4809 C012 003C 00F0          A2 EQL 0             % SET UP LOGICAL TEST FOR BLANK ADDR      0C333C00 D
0CB4  580B 00C0 003C 00F0          IF TRUE THEN STEP ELSE SKIP                                    00334C00 D
0CB5  010C 0000 0030 0040          INLINE - 1 = MPCR    %IF ADDR = 0 ADD DATA IN NEXT ADDR        0C335C00 D
00B6  4809 C0C0 001C 00F0          A2 = MAR2            % PUT MEMORY ADDR IN MAR2 FOR WRITE OP    0C336C00 D
0CB7  05E0 000D 0030 0040          OUT - 1 = MPCR       % WRITE DATA IN MEMORY AND GET NEXT CARD 0C337C00 D
                                                        %                                        00338000 D
                                                        %                                        00339C00 D
                           INLINE:                      % THIS ROUTINE WILL INSERT DATA INTO THE 0C340000 D
                                                        % ADDRESS CONTAINED IN THE NXTINSTR       00341C00 D
                                                        % REG.  THE DATA TO BE WRITTEN IS IN MIR.0C342C00 D
0CB8  4809 200D 201C 00F0          LIT = MAR2,A2        % PUT ADDR OF NEXTINST INTO MAR2, A2      00343C00 D
0CB9  0230 00C0 0030 00E0          NXTINSTR = LIT                                                 00344C00 D
0CBA  0460 00C0 0030 0060          INPUT - 1 = CPCR     % READ CONTENTS OF NXTINSTR              00345C00 D
0CBB  4809 0C40 001C 00F0          B = MAR2             % PLACE CONTENTS OF NXTINSTR INTO MAR2    00346C00 D
0CBC  0810 00C0 0C00 0060          OUTPUT1 - 1 = CPCR   % PLACE DATA AT THAT ADDRESS(MAR2)       00347C00 D
0CBD  4809 0C46 0030 00F0          B + 1 = MIR          % CREATE NEXT ADDR FOR NXTINSTR           00348C00 D
0CBE  4809 C0C0 001C 00F0          A2 = MAR2            % WRITE NEW ADDR IN NXTINSTR              00349C00 D
0CBF  05E0 00C0 0C00 0040          OUT - 1 = MPCR       % REG. AND GET THE NEXT CARD              00350C00 D
                                                        %                                        00351000 D
                                                        %                                        00352C00 D
                           SETINDEXREG:                 % THIS ROUTINE WILL EXAMINE COL. 2 FOR    00353C00 D
                                                        % THE GENERAL REGISTER STACK NO. AND      00354000 D
                                                        % COL. 7-8 FOR THE REGISTER NO.  THE      00355000 D
                                                        % CONTENTS OF COL. 13-20(ONLY 16-20 USED)00356C00 D
                                                        % WILL THEN BE CONVERTED TO OCTAL AND     0C357000 D
```

```
                                                        % STORED IN THE DESIRED GENERAL REGISTER.00358000 D
00C0  0460 0000 0000 0060          INPUT - 1 = CPCR     % REREAD COL. 1 - 4                      00359000 D
00C1  4809 0C41 0B30 00F0          B L = B              % ISOLATE COLUMN 2                       00360000 D
00C2  0000 0000 0030 0030          COMP 8 = SAR                                                   00361000 D
00C3  4809 0C40 C800 00F0          B R = B,A1                                                     00362000 D
00C4  4809 A0DE 0B30 00F0          A1 - 1 = B           % B IS NOW O(GEN REG. NO. 1) OR 1(NO. 2) 00363000 D
00C5  4809 0C41 0B30 00F0          B L = B              % ADDRESS OF GENERAL REGISTER STACK      00364000 D
00C6  8000 0000 0030 0030          COMP 4 = SAR         % MULTIPLY BY 16                        00365000 D
00C7  4809 0C41 0028 00F0          B L = BR1,LMAR       % STORE BASE ADDR OF GEN REG STACK       00366000 D
00C8  0000 0000 0030 0030          COMP 8 = SAR         % SHIFT TO PROPER POS. FOR BR1           00367000 D
00C9  4809 20C0 001C 00F0          LIT = MAR2           % LOAD ADDR OF CARD COL 5-8              00368000 D
00CA  0780 0000 0000 00E0          CARD5X8 = LIT                                                  00369000 D
00CB  0460 0000 0000 0060          INPUT - 1 = CPCR     % READ CARD COL. 5-8                    00370000 D
00CC  4809 0C41 0B30 00F0          B L = B              % ISOLATE CARD COL 7-8                   00371000 D
00CD  0000 0000 0030 0020          COMP 16 = SAR                                                  00372000 D
00CE  4809 0C40 C000 00F0          B R = A1             % A1 CONTAINS COL 7-8(GEN REG NO.)       00373000 D
00CF  0880 0000 0000 0060          PACKED = CPCR        % JUMP TO 2ND LINE IN ROUTINE(4 DIGIT NO)00374000 D
00D0  0970 0000 0000 0060          DECOCT - 1 = CPCR    % B HAS OCTAL VALUE OF GEN REG NUMBER    00375000 D
00D1  4809 0000 0030 20F0          ASR                  % SET MAR1 AS MOST RECENTLY REFERENCED   00376000 D
00D2  4809 0F40 C000 00F0          BMAR R = A1          % MOVE BASE ADDR TO A1 FROM BR1          00377000 D
00D3  0000 0000 0030 0010          B = SAR              % SHIFT BMAR TO ISOLATE BR1              00378000 D
00D4  4809 AC41 0020 00F0          A1 + B L = BR1       % CREATE GEN REG ADDR(BASE + OFFSET)     00379000 D
00D5  0700 0000 0030 00B0          COMP 8 = SAR; CARD13X16 = LIT  % SHIFT FOR BR1                00380000 D
00D6  4809 20C0 001C 00F0          LIT = MAR2           % ADDRESS OF DATA                        00381000 D
00D7  0460 0000 0000 0060          INPUT - 1 = CPCR     % READ IN DATA                           00382000 D
00D8  0510 0000 0030 0060          GETADDR - 1 = CPCR                                             00383000 D
00D9  4809 0000 0030 20F0          ASR                  % SET MAR1 AS MOST RECENTLY REFERENCED   00384000 D
00DA  4809 0F40 801C 00F0          BMAR R = MAR2        % LOAD GEN REG ADDR INTO MAR2            00385000 D
00DB  0000 0000 0030 C010          B = SAR              % SHIFT TO ISOLATE BR1                   00386000 D
00DC  4809 0C40 0030 00F0          B = MIR              % B HAS OCTAL VALUE OF DATA              00387000 D
00DD  05E0 0000 0000 C040          OUT-1 = MPCR         %WRITE DATA FROM COL. 13-20 INTO        00388000 D
                                                        %SPECIFIED GENERAL REGISTER             00389000 D
                                                        %                                       00390000 D
                                                        %                                       00391000 D
                                   BREAKPOINT:          %SET UP BREAKPOINT REGISTER AND SET     00392000 D
                                                        % STATUS BITS                           00393000 D
00DE  4809 0000 0000 00F0          BEX                  % FILL B WITH CARD COL 1-4   (FROM SWI) 00394000 D
00DF  4809 0C41 4030 00F0          B L = A1             % ISOLATE THE R,W,B CHARACTER            00395000 D
00E0  0000 0000 0030 0030          COMP 8 = SAR                                                   00396000 D
00E1  4809 A0C0 C030 00F0          A1 R = A1            %ISOLATE COLUMN 2                       00397000 D
00E2  0000 0000 0030 0030          24 = SAR                                                       00398000 D
00E3  4809 20C0 001C 00F0          LIT = MAR2           % WRITE PSW INTO B                      00399000 D
00E4  0200 0000 0030 00E0          PSW = LIT                                                      00400000 D
00E5  0460 0000 0000 0060          INPUT - 1 = CPCR     % INPUT FROM MEMORY                     00401000 D
00E6  4809 A152 0030 00F0          A1 EQL LIT           %CHECK FOR CHARACTER :R:                00402000 D
00E7  0290 0000 0030 00E0          41 = LIT                                                       00403000 D
00E8  6419 A152 0030 00F0          IF FALSE THEN A1 EQL LIT: SKIP                                 00404000 D
00E9  01E0 0000 0000 C040          BRKREAD - 1 = MPCR   %JUMP TO  BREAKPOINT READ SUBFUNCTION     00405000 D
00EA  0360 0000 0030 00E0          54 = LIT             %CHECK FOR :W: CHARACTER                  00406000 D
00EB  6019 0000 0030 00F0          IF FALSE THEN SKIP                                             00407000 D
00EC  01F0 0000 0000 0040          BRKWRITE - 1 = MPCR  % JUMP TO BREAKPOINT WRITE SUBFUNCTION    00408000 D
00ED  0200 0000 0030 0040          BRKBOTH -1 = MPCR    % DEFAULTS TO BREAKPOINT BOTH (R+W)      00409000 D
                                                        %                                       00410000 D
                                                        %                                       00411000 D
                                   BRKREAD:             % THIS ROUTINE SETS UP THE BREAKPOINT    00412000 D
                                                        % REGISTER AND SETS THE READ STATUS BITS 00413000 D
00EE  4809 00C1 4030 00F0          1 L = A1                                                       00414000 D
00EF  8000 0000 0030 0010          COMP 20 = SAR                                                  00415000 D
00F0  4809 AC30 0030 00F0          A1 OR B = MIR        % SET BIT # 20 OF PSW REG                00416000 D
00F1  0810 0000 0000 0060          OUTPUT1 - 1 = CPCR   % WRITE PSW INTO MEMORY                 00417000 D
```

```
00F2  021C 0000 0030 0040          WRITEBRKPT - 1 = MPCR % JUMP TO WRITE BREAKPOINT REGISTER      00418C00 D
                                                         %                                         00419C00 D
                                BRKWRITE:                % THIS ROUTINE SETS UP THE BREAKPOINT    00420C00 D
                                                         % REGISTER AND SETS THE WRITE STATUS BITS00421000 D
00F3  4809 0001 4030 00F0          1 L = A1                                                        00422C00 D
00F4  3000 0000 0000 C010          COMP 21 = SAR                                                   00423C00 D
00F5  4809 AC5C 0030 00F0          A1 OR B = MIR         %SET BIT 21 OF PSW REG                    00424C00 D
00F6  0810 0000 0030 0060          OUTPUT1 - 1 = CPCR    % WRITE PSW INTO MEMORY                 00425C00 D
00F7  022C 0000 0030 0040          WRITEBRKPT - 1 = MPCR % JUMP TO WRITE BREAKPOINT REGISTER     00426C00 D
                                                         %                                         00427C00 D
                                BRKBOTH:                 % THIS ROUTINE  SETS UP THE BREAKPOINT   00428C00 D
                                                         % REGISTER AND SETS THE STATUS BITS FOR  00429C00 D
                                                         % BOTH READ AND WRITE                    00430C00 D
00F8  4809 2001 4030 00F0          LIT L = A1                                                      00431C00 D
00F9  030C 0000 0000 00A0          COMP 16 = SAR; 48 = LIT  % SET UP A1 WITH MASK BIT POS. 21/20   00432C00 D
00FA  4809 AC5C 0030 00F0          B OR A1 = MIR         % PUT BREAKPOINT CODE INTO SR1 BIT POS   00433C00 D
                                                         % 21 AND 20                              00434C00 D
00FB  081C 00C0 0000 0060          OUTPUT1 - 1 = CPCR    % WRITE PSW INTO MEMORY                 00435C00 D
                                                         %                                         00436C00 D
                                                         %                                         00437C00 D
                                WRITEBRKPT:              %THIS ROUTINE WRITES THE DECIMAL ADDR     00438C00 D
                                                         % INTO THE BREAKPOINT REGISTER AND       00439C00 D
                                                         % AND INTO  REGISTER BR1                 00440C00 D
00FC  4809 2003 001C 00F0          LIT = MAR2            % RETRIEVE CARD1X4 CONTENTS              00441C00 D
00FD  07A0 0000 0030 00E0          CARD1X4 = LIT                                                   00442C00 D
00FE  0460 00C0 0030 0060          INPUT - 1 = CPCR      % INPUT INTO B                          00443C00 D
00FF  0510 0000 0000 0060          GETADDR - 1 = CPCR    % OBTAIN ADDRESS FROM COL 4-8            00444C00 D
0100  4809 2003 001C 00F0          LIT = MAR2                                                      00445C00 D
0101  021C 00C0 0030 00E0          BKPT = LIT            % GET ADDRESS OF BKPT REG                00446C00 D
0102  4809 0C40 003C 00F0          B = MIR               % B HAS BINARY  ADDRESS VALUE            00447C00 D
0103  4809 0C41 0020 00F0          B L = BR1             %INSERT BREAKPOINT ADDR INTO BR1          00448C00 D
0104  0000 0000 0030 0030          COMP 8 = SAR                                                    00449C00 D
0105  0810 0000 0030 0060          OUTPUT1 - 1 = CPCR    % WRITE OUT ADDR INTO BKPT REG          00450C00 D
0106  060C 0000 0000 0040          NEWCARD - 1 = MPCR    % RETURN TO NEW CARD ROUTINE            00451C00 D
                                                         %                                         00452C00 D
                                                         %                                         00453C00 D
                                DOUBLEWORD:              %HANDLES SECOND WORD OF TWO WORD INSTR    00454C00 D
0107  4809 E000 8800 00FC          A3 R = B              % PUT COL 16 IN LSB                      00455C00 D
0108  07D0 0000 0000 00A0          16 = SAR; CARD13X16 = LIT                                       00456C00 D
0109  4809 2003 001C 00F0          LIT = MAR2            %RELOAD ADDR OF CARD COL 13-16            00457C00 D
010A  052C 0000 0000 0060          GETADDR = CPCR        %OMIT BEX STEP IN GETADDR, B CONTAINS     00458C00 D
010B  4809 0C40 203C 00F0          B = A2                % COL 16 DIGIT IN LSB OF B                00459C00 D
010C  023C 00C0 0030 0060          LOADINSTR - 1 = CPCR  %LOAD OCTAL VALUE OF DATA INTO SECOND     00460C00 D
010D  060C 0000 003C 0040          NEWCARD - 1 = MPCR    % WORD OF DOUBLEWORD INSTRUCTION AND     00461C00 D
                                                         %                                         00462C00 D
                                                         %                                         00463C00 D
                                INSTRUCTION:             % THIS ROUTINE WILL PACK AN INSTRUCTION   00464C00 D
                                                         % INPUT CARD INTO A 16 BIT INSTRUCTION   00465C00 D
                                                         % AND PLACE IT IN THE LS 16 BITS OF A    00466C00 D
                                                         % 32 BIT WORD.  RITYPE1 INSTRUCTIONS ARE 00467C00 D
                                                         % ARE SELECTED  AND HANDLED BY A SEPARATE00468C00 D
                                                         % ROUTINE.                               00469C00 D
010E  4809 2003 001C 00F0          LIT = MAR2                                                      00470C00 D
010F  97B0 00C0 0000 00A0          CARD5X8 = LIT; 21 = SAR                                         00471C00 D
0110  0460 C000 0000 0060          INPUT - 1 = CPCR      % READ THE CONTENTS OF COL. 5-8 INTO B   00472C00 D
0111  4809 0C40 A030 00F0          B R = A2              % ISOLATE 11 BITS (CHARACTER + 3 BITS)   00473C00 D
0112  4809 0C41 1030 00F0          B L = A3              % ISOLATE NEXT OCTAL DIGIT               00474C00 D
0113  3000 0000 003C 002C          COMP 13 = SAR                                                   00475C00 D
0114  4809 E000 9030 00F0          A3 R = A3                                                       00476C00 D
0115  9000 0000 0030 0030          29 = SAR                                                        00477C00 D
```

```
0116  4809 C000 0090 00F0          A2 = AMPCR          % SET UP A Y-SELECT                    00478000 D
0117  4809 E641 0090 00F0          A3 + AMPCR L = AMPCR % PACK OPCODE IN SIX BITS            00479000 D
0118  A000 0000 0030 0030          COMP 2 = SAR        % IN A2 AND SHIFT 2 BITS FOR NEXT FIELD 00480000 D
0119  4809 0C41 1030 40F0          B L = A3, CSAR      % ISOLATE FUNCTION BITS                00481000 D
011A  27C0 C000 0000 C080          COMP 30 = SAR; CARD9X12 = LIT                             00482000 D
011B  4809 E0C0 9030 00F0          A3 R = A3                                                 00483000 D
011C  4809 E640 2030 00F0          A3 + AMPCR = A2     % PACK FUNCTION BITS (0-3) INTO A2     00484000 D
011D  4809 2000 001C 00F0          LIT = MAR2                                                00485000 D
011E  0460 0000 0090 0060          INPUT - 1 = CPCR    % READ IN NEXT 4 COL. (9-12)          00486000 D
011F  4809 2C55 9090 C0F0          B AND LIT = A1                                            00487000 D
0120  0FFC 0000 0000 00E0          255 = LIT           % ISOLATE COL. 12 TO CHECK FOR COMMA   00488000 D
0121  4809 A152 0030 00FC          A1 EQL LIT                                                00489000 D
0122  03A0 0000 0030 C0E0          58 = LIT            % CODE FOR COMMA                      00490000 D
0123  500B 0000 0000 C0F0          IF FALSE THEN STEP ELSE SKIP % TREAT COL. 10-12 AS "D" FIELD 00491000 D
                                                       % OF RITYPE 1 INSTRUCTION             00492000 D
0124  0240 0000 0030 0040          RITYPE1 - 1 = MPCR  % TRANSFER TO RITYPE1 ROUTINE         00493000 D
                                                       % CARD HAS COL. 10-11 AS "A" FIELD    00494000 D
                                                       % COL. 13-14 AS "M" FIELD             00495000 D
0125  4809 0C41 0B0C C0F0          B L = B             %ISOLATE CC 10,11                     00496000 D
0126  1000 0000 0030 0020          COMP 15 = SAR                                             00497000 D
0127  4809 0C40 C090 C0F0          B R = A1            % ISOLATE 1 BIT OF CC 10 + 3 BITS      00498000 D
0128  8000 00C0 0030 0030          28 = SAR                                                  00499000 D
0129  4809 0C40 8B00 C0F0          B R = B                                                   00500000 D
012A  B070 0000 0000 00A0          23 = SAR; 7 = LIT   % ISOLATE CC 11 (3BITS) INTO B        00501000 D
012B  4809 2C56 0B90 00F0          B AND LIT = B                                             00502000 D
012C  4809 AC40 0040 C0F0          A1 + B = AMPCR      % INSERT "A" FIELD INTO AMPCR          00503000 D
012D  4809 C0C1 2090 00F0          A2 L = A2                                                 00504000 D
012E  970C 0000 0000 00B0          COMP 4 = SAR; CARD13X16 = LIT                             00505000 D
012F  4809 C641 2030 00F0          A2 + AMPCR L = A2   % 4 BITS FOR :M: FIELD                00506000 D
0130  4809 2000 001C C0F0          LIT = MAR2                                                00507000 D
0131  0460 0000 0030 C06C          INPUT - 1 = CPCR                                          00508000 D
0132  4809 0C41 4B30 C0F0          B L = A1, B         %ISOLATE BIT WHICH IS COL 13 DIGIT(1,0) 00509000 D
0133  100C 0000 0030 C030          COMP 7 = SAR                                              00510000 D
0134  4809 A0C0 C090 00F0          A1 R = A1           %FILL A1 WITH EIT-0-0-0 (LSB)         00511000 D
0135  9000 00C0 0030 0030          28 = SAR                                                  00512000 D
0136  4809 0C41 1B30 C0F0          B L = A3,B          %PUT COL14 IN LSB OF A3,B             00513000 D
0137  2000 00C0 0030 0030          COMP 6 = SAR                                              00514000 D
0138  4809 0C40 8B90 00F0          B R = B             %PUT COL14 IN LSB OF B                00515000 D
0139  900C 00C0 0030 0030          29 = SAR                                                  00516000 D
013A  4809 AC40 0B90 C0F0          A1 + B = B          % COMBINE COL. 13 AND 14              00517000 D
013B  4809 CC40 2000 C0F0          A2 + B = A2         % FINISH BUILDING 16 BIT INSTR         00518000 D
013C  0250 00C0 0000 006C          LOADINSTR - 1 = CPCR % LOAD INSTRUCTION(A3 UNCHANGED)     00519000 D
013D  4809 E001 1090 C0F0          A3 L = A3           % A3 CONTAINS COLS. 15-16 IN UPPER HALF 00520000 D
013E  9000 00C0 C090 0030          COMP 3 = SAR                                              00521000 D
013F  4809 E000 C090 C0F0          A3 R = A1           % SET UP COL. 15 IN LSB OF A1          00522000 D
0140  03A0 00C0 0030 00B0          24 = SAR; 58 = LIT                                        00523000 D
0141  4809 A152 0030 00FC          A1 EQL LIT          %CHECK FOR COMMA(,)                   00524000 D
0142  680B 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP                               00525000 D
0143  106C 00C0 C090 0040          DOUBLEWORD - 1 = MPCR  %COLS. 16-?- OF CARD CONTAIN DECIMAL 00526000 D
                                                       % :Y: FIELD                           00527000 D
0144  060C 00C0 0090 0C4C          NEWCARD - 1 = MPCR  % GET NEW CARD                        00528000 D
                                                       %                                     00529000 D
                                                       %                                     00530000 D
                                LOADINSTR:             % THIS ROUTINE WILL LOAD AN INSTRUCTION 00531000 D
                                                       % WHICH IS CONTAINED IN A2             00532000 D
                                                       % INTO THE NEXT APPROPRIATE S-MEMORY WORD00533000 D
0145  4809 0640 4090 C0F0          AMPCR = A1          % A1 IS A TEMP RETURN ADDRESS REG      00534000 D
0146  4809 2000 001C 00F0          LIT = MAR2                                                00535000 D
0147  0230 0000 0030 00E0          NXTINSTR = LIT      % READ NEXT LOAD ADDRESS FROM NXTINSTR 00536000 D
0148  4809 0000 0071 00F0          LCTR                % SAVE NXTINSTR IN CTR (COMPLEMENTED)  00537000 D
```

```
0149  0460 0000 0000 0060          INPUT - 1 = CPCR                          00538000 D
014A  4809 0C43 C01C 00FC          B = MAR2                  % PLACE LOAD ADDRESS INTO MAR2   00539000 D
014B  4809 0C46 0030 00F0          B + 1 = B                 % INCREMENT LOAD ADDRESS         00540000 D
014C  4809 C000 0C50 00F0          A2 = MIR                  % TRANSFER INSTRUCTION WORD IN A2 TO MIR 00541000 D
014D  0810 C0C3 0030 0060          OUTPUT1 - 1 = CPCR        % AND WRITE INTO S-MEMORY        00542000 D
014E  4809 C802 801C 00F0          NOT CTR R = MAR2          % WRITE LOAD ADDRESS             00543000 D
014F  3000 0000 0030 0030          29 = SAR                  % INTO NXTINSTR                  00544000 D
0150  4809 0C49 0030 00F0          B = MIR                                                    00545000 D
0151  0810 0000 0030 0060          OUTPUT1 - 1 = CPCR        % MEMORY WRITE                   00546000 D
0152  4809 A0C3 0030 00F0          A1 = AMPCR                % REINSTALL RETURN ADDRESS        00547000 D
0153  4809 0000 0030 00F0          STEP                      % REQUIRED BEFORE JUMP           00548000 D
0154  4820 0003 0030 00F0          JUMP                                                       00549000 D
                                                             %                                00550000 D
                                                             %                                00551000 D
                        MACHSTAT:                            % THIS ROUTINE WILL DISPLAY       00552000 D
                                                             % THE AN/UYK-20 REG CONTENTS.     00553000 D
                                                             % IF M1  CONTROL CARD IS USED, THE 00554000 D
                                                             % OUTPUT WILL BE DISPLAYED ON THE CRT. 00555000 D
                                                             % THE SWI CONTAINS CC COL. 1-4.   00556000 D
0155  4809 2003 0C1C 00F0          LIT = MAR2, BEX           % LOAD ADDR OF SR#2 INTO MAR2     00557000 D
0156  0220 0000 0000 00E0          STATUS2 = LIT                                              00558000 D
0157  4809 0C40 8800 00F0          B R = B                   % CC COL. 2 INTO LS BYTE OF B     00559000 D
0158  03F0 0003 0030 00A0          16 = SAR; 63 = LIT                                         00560000 D
0159  4809 2055 2000 00F0          LIT AND B = A2            % MASK OFF COL. 2                00561000 D
015A  1000 0003 0030 0000          1 = SAR                                                    00562000 D
015B  4809 C002 0030 00FC          A2 EQL 1                  % CHECK FOR M1                   00563000 D
015C  5C19 1801 C030 00F0          IF TRUE THEN B10C C = A1; SKIP  % SET BIT 30 OF SR#2 FOR CRT 00564000 D
015D  4809 0000 4030 00FC          0 = A1                                                     00565000 D
015E  0460 0003 0000 0060          INPUT - 1 = CPCR          % READ IN CONTENTS OF SR#2 INTO B 00566000 D
015F  4809 AC5C 0C30 00F0          A1 OR B = MIR             % SET APPROPRIATE SR#2 BITS      00567000 D
0160  0810 0003 0030 0060          OUTPUT1 - 1 = CPCR        % WRITE NEW SR#2                 00568000 D
0161  0260 0003 0030 0060          DUMPREG - 1 = CPCR        % DISPLAY REG CONTENTS           00569000 D
0162  0600 0000 0030 0040          NEWCARD - 1 = MPCR        % FETCH NEXT CARD                 00570000 D
                                                             %                                00571000 D
                                                             %                                00572000 D
                        RESERVESPACE:                        % THIS ROUTINE WILL RESERVE A DECIMAL 00573000 D
                                                             % NUMBER OF MEMORY CELLS AS INDICATED BY 00574000 D
                                                             % COL. 4-8.                       00575000 D
0163  0510 0003 0030 0060          GETADDR - 1 = CPCR        % GET NUMBER OF CELLS AND CONVERT TO OCT.00576000 D
0164  4809 0C40 4000 00F0          B = A1                    % A1 CONTAINS NUMBER OF CELLS BEING SAVED00577000 D
0165  4809 2003 001C 00F0          LIT = MAR2                                                 00578000 D
0166  0230 00C0 000C 00E0          NXTINSTR = LIT                                             00579000 D
0167  0460 0003 0030 0060          INPUT - 1 = CPCR          % READ PRESENT NXTINSTR REG CONTENTS 00580000 D
0168  4809 AC40 0030 00F0          A1 + B = MIR              % NEXT INSTRUCTION ADDR TO MIR    00581000 D
0169  05E0 0003 0000 0040          OUT - 1 = MPCR            % WRITE OUT A NEW NEXT INST LOCATION 00582000 D
                                                             % INTO THE NXINSTR REG           00583000 D
                                                             %                                00584000 D
                                                             %                                00585000 D
                        RITYPE1:                             % THIS ROUTINE PACKS THE "C" FIELD INTO 00586000 D
                                                             % A2 (D FIELD IN CARD COL. 10-12) 00587000 D
016A  4809 0C40 9000 00F0          B R = A3                  % COL 10 TO LS 3 BITS OF A3       00588000 D
016B  0070 0003 0030 00A0          16 = SAR; 7 = LIT                                          00589000 D
016C  4809 E157 1000 00FC          A3 AND LIT L = A3         % ISOLATE COL 10 + 3 BITS INTO LSB OF A3 00590000 D
016D  3000 0003 0030 0030          COMP 3 = SAR                                               00591000 D
016E  4809 0C40 C000 00F0          B R = A1                  % COL 11 TO LSB OF A1            00592000 D
016F  0000 00C0 0030 C010          B = SAR                                                    00593000 D
0170  4809 A155 0030 00FC          A1 AND LIT = AMPCR        % ISOLATE COL 11 INTO AMPCR      00594000 D
0171  4809 E650 1000 00F0          A3 OR AMPCR L = A3        % A3 = COLS. 10,11, + 3 BLANKS    00595000 D
0172  3000 0003 0030 003C          COMP 3 = SAR                                               00596000 D
0173  4809 2056 0B30 00F0          LIT AND B = B             % ISOLATE COL. 12                00597000 D
```

```
0174  4809 EC52 0B0C 00F0          A3 OR B = B          % B= 0000/C0+COL. 10,11,12          00598000 D
0175  4809 C001 2C00 00F0          A2 L = A2            % PREPARE A2 FOR ADDITION OF B      00599000 D
0176  0000 0003 0030 0030          COMP 8 = SAR                                              00600000 D
0177  4809 CC40 2000 00F0          A2 + B = A2          % PACK REMAINING BITS INTO A2       00601000 D
0178  1440 0000 0000 CC60          LOADINSTR - 1 = CPCR % INSTRUCTION WORD RITYPE 1 COMPLETED 00602000 D
0179  05E0 0000 0C00 C040          OUT - 1 = MPCR                                            00603000 D
                                                        %                                    00604000 D
                                                        %                                    00605000 D
                                   SETPAGING:           % THIS ROUTINE WILL SET THE PAGING BIT 00606000 D
                                                        % (BIT POS. 22) OF THE PSW          00607000 D
017A  4809 2003 001C 00F0          LIT = MAR2                                                00608000 D
017B  A200 0000 0000 00A0          PSW = LIT, 22 = SAR                                       00609000 D
017C  046C 0003 0000 0060          INPUT - 1 = CPCR     % READ IN CONTENTS OF PSW           00610000 D
017D  4809 0041 8030 40F0          B C = B, CSAR        % MOVE PAGING BIT TO LS BIT          00611000 D
017E  4809 2050 8800 00FC          LIT OR B C = B       % PAGING BIT SET(PAGING SELECTED)    00612000 D
017F  0010 0000 0000 C0E0          1 = LIT                                                   00613000 D
0180  4809 0040 0030 00F0          B = MIR              % CONTENTS OF PSW INTO MIR           00614000 D
0181  05E0 0000 0C00 0040          OUT - 1 = MPCR                                            00615000 D
                                                        %                                    00616000 D
                                                        %                                    00617000 D
                                   CHARSTRING:          % THIS ROUTINE WILL READ THE CHARACTER 00618000 D
                                                        % STRING STARTING IN COL. 9 UNTIL FINDING00619000 D
                                                        % A QUOTE (*).  THE STRING WILL THEN BE 00620000 D
                                                        % STORED AT THE ADDRESS IN COL. 4-8.  IF 00621000 D
                                                        % COL. 4-8 IS BLANK, THE STORAGE ADDRESS 00622000 D
                                                        % WILL DEFAULT TO THE ADDRESS IN NXTINSTR00623000 D
0182  2809 0000 0000 00F0          IF LC1               % CLEAR CONDITION BITS               00624000 D
0183  3809 0000 0000 00F0          IF LC2                                                    00625000 D
0184  0510 0000 0030 0060          GETADDR - 1 = CPCR   % GET ADDRESS IN COL. 4-8           00626000 D
0185  4809 2003 001C 00F0          LIT = MAR2           % MAR2 CONTAINS ADDR OF NEXT INSTRUCTION 00627000 D
0186  0230 0000 0030 00E0          NXTINSTR = LIT                                            00628000 D
0187  4809 0052 0030 00F0          B EQL 0              % CHECK FOR BLANK ADDRESS FIELD      00629000 D
0188  6A46 0000 0030 00F0          IF TRUE THEN SET LC2; STEP ELSE SKIP                      00630000 D
0189  0460 0000 0000 0060          INPUT - 1 = CPCR     % READ CONTENTS OF NXTINSTR          00631000 D
018A  4809 0040 4030 00F0          B = A1               % A1 CONTAINS ADDR TO BE WRITTEN INTO 00632000 D
018B  4809 2003 001C 00F0          LIT = MAR2           % SET UP READ ADDR OF COL. 9-12 IN MAR2 00633000 D
018C  0700 0000 0030 00EC          CARD9X12 = LIT                                            00634000 D
018D  046C 0000 0030 0060          MORECHAR:  INPUT - 1 = CPCR  % READ A 4 CHARACTER BLOCK  00635000 D
018E  0030 0000 0000 00EC          3 = LIT                                                   00636000 D
018F  4812 0003 0071 00F0          LCTR; SAVE           % SET UP COUNTER FOR 4 ITERATIONS    00637000 D
0190  4809 0041 8032 00F0          B C = B, MIR, INC    % SHIFT 1 CHARACTER INTO B TO BE ANALYZED00638000 D
0191  0000 0000 0000 0010          8 = SAR                                                   00639000 D
0192  4809 2056 2000 00F0          LIT AND B = A2       % EXTRACT RIGHTMOST CHARACTER        00640000 D
0193  03F0 0003 0030 00E0          63 = LIT             % SET UP 6 BIT MASK                  00641000 D
0194  4809 0152 0030 00F0          A2 EQL LIT                                                00642000 D
0195  00F0 0000 0030 00E0          15=  LIT             % TEST FOR * (OF)                    00643000 D
0196  6BC9 0003 0030 00F0          IF TRUE THEN SET LC1   % IF A (*) IS IN THE STRING SET LC1 00644000 D
0197  0210 0000 0030 00E0          33 = LIT             % ADD VALUE TO (*) SO THAT SPACE APPEARS 00645000 D
0198  2FC9 2040 0B30 00F0          IF LC1 THEN LIT + B = B,MIR; SET LC1                      00646000 D
0199  8029 0003 0030 00F0          IF NOT COV THEN JUMP  % CHECK IF LOOP DONE 4 TIMES        00647000 D
019A  4809 0F46 1030 00F0          BMAR + 1 = A3        % SET UP NEXT READ ADDRESS           00648000 D
019B  4809 A003 001C 00F0          A1 = MAR2            % MAR2 NOW HAS WRITE ADDRESS         00649000 D
019C  0810 0000 0030 0060          OUTPUT1 - 1 = CPCR   % WRITE 4 CHAR BLOCK TO MEMORY       00650000 D
019D  4809 A000 4030 00F0          A1 + 1 = A1          % INCREMENT WRITE ADDRESS            00651000 D
019E  28CB 0003 0030 00F0          IF LC1 THEN STEP ELSE SKIP  % CHECK IF (*) WAS FOUND      00652000 D
019F  027C 0000 0000 0040          ENDCHARSTRING - 1 = MPCR                                  00653000 D
01A0  4809 E003 001C 00F0          A3 = MAR2            % A3 AND MAR2 HAVE READ ADDRESS      00654000 D
01A1  4809 E159 0030 00F0          A3 LEQ LIT           % CHECK IF YOU ARE AT THE END OF THE CARD00655000 D
01A2  0800 0003 0030 00E0          CARD77X80 = LIT                                           00656000 D
01A3  700B 0000 0000 00FC          IF TRUE THEN STEP ELSE SKIP                               00657000 D
```

```
01A4  1800 0000 0030 0040          MORECHAR - 1 = MPCR     % READ IN MORE CHARACTERS                00658000 D
01A5  3008 0000 0030 00F0   ENOCHARSTRING:  IF NOT LC2 THEN STEP ELSE SKIP                          00659000 D
01A6  0600 0000 0030 0040          NEWCARD - 1 = MPCR      % GET NEXT CARD                          00660000 D
01A7  4809 A000 0030 00F0          A1 = MIR                % STORE A1 CONTENTS INTO MIR              00661000 D
01A8  4809 2003 001C 00F0          LIT = MAR2              % LOAD MAR2 WITH ADDR OF NXTINSTR         00662000 D
01A9  0230 0000 0030 00E0          NXTINSTR = LIT                                                    00663000 D
01AA  05E0 0000 0000 0040          OUT - 1 = MPCR          % UPDATE CONTENTS OF NXTINSTR             00664000 D
                                                           %                                         00665000 D
                                                           %                                         00666000 D
                            SWITCHES:                      % THIS ROUTINE WILL EMULATE THE PHYSICAL   00667000 D
                                                           % SETTING OF THE UYK-20 PROGRAM STOP       00668000 D
                                                           % SWITCHES 1 AND 2.  MAR2 CONTAINS ADDR    00669000 D
                                                           % ADDRESS OF COL 1-9.  COL 2 WILL CONTAIN  00670000 D
                                                           % A 1(INDICATES SW#1 ON) OR A 2(INDICATES  00671000 D
                                                           % SW#2 IS ON).  BITS 30 AND 29 OF THE      00672000 D
                                                           % PSW WILL BE SET TO A "1" TO INDICATE     00673000 D
                                                           % SWITCH 1 AND SWITCH 2 RESPECTIVELY.      00674000 D
01AB  0460 0000 0030 0060          INPUT - 1 = CPCR        % REREAD COL. 1-9 INTO B                  00675000 D
01AC  4809 0C40 C000 00F0          B R = A1                % COL. 2 INTO LS BYTE  OF A1               00676000 D
01AD  03F0 0000 0000 00A0          16 = SAR; 63 = LIT                                                00677000 D
01AE  4809 A156 9030 00F0          A1 AND LIT = A1         % MASK OUT LS BYTE OF A1                  00678000 D
01AF  4809 2003 001C 00F0          LIT = MAR2              % MAR2 CONTAINS ADDR OF PSW               00679000 D
01B0  0200 0000 0030 00E0          PSW = LIT                                                         00680000 D
01B1  0460 0000 0030 C060          INPUT - 1 = CPCR        % READ INTO B THE CONTENTS OF THE PSW     00681000 D
01B2  4809 A152 0030 00F0          A1 EQL LIT              % CHECK IF SW#1 SET                       00682000 D
01B3  0010 0000 0000 00EC          1 = LIT                                                           00683000 D
01B4  6808 0000 0030 00F0          IF TRUE THEN STEP ELSE SKIP                                       00684000 D
01B5  0280 0000 000C 0040          SETCOL30 - 1 = MPCR     % SET COL 30                             00685000 D
01B6  4809 0C41 8830 40F0          B C = B, CSAR           % SET COL 29(DEFAULTS TO SW#2 BEING SET)  00686000 D
01B7  9010 0000 0030 00B0          29 = SAR; 1 = LIT                                                 00687000 D
01B8  4809 2050 0830 00F0          LIT OR B = B            % SET BIT 29                             00688000 D
01B9  4809 0C41 8830 00F0          B C = B, MIR            % RESTORE B                               00689000 D
01BA  05E0 0000 0000 0040          OUT - 1 = MPCR                                                    00690000 D
01BB  4809 0C41 8830 40F0   SETCOL30:  B C = B, CSAR                                                 00691000 D
01BC  A010 0000 0000 00B0          30 = SAR; 1 = LIT                                                 00692000 D
01BD  4809 2050 0830 00F0          LIT OR B = B            % SET BIT 30                             00693000 D
01BE  4809 0C41 8030 00F0          B C = MIR               % RESTORE B VALUE IN MIR                  00694000 D
01BF  05E0 0000 0030 0040          OUT - 1 = MPCR                                                    00695000 D
                                                           %                                         00696000 D
                                                           %                                         00697000 D
01C0  4809 2003 001C 00F0   TRACE:  LIT = MAR2, BEX        % THIS ROUTINE SETS THE 2 MS BITS IN SR#2  00698000 D
01C1  0220 0000 0030 00E0          STATUS2 = LIT           % FOR DUMP REGISTER CALLS TO CRT OR PTR   00699000 D
01C2  4809 0C40 8830 00F0          B R = B                 % ISOLATE COL. 2 OF CONTROL CARD          00700000 D
01C3  03F0 0000 0000 00A0          16 = SAR; 63 = LIT      % IF 1 THEN SET MS 2 BITS OF SR#2 (CRT)   00701000 D
01C4  4809 2056 2030 00F0          LIT AND B = A2          % A2 CONTAINS COL. 2 OF CONTROL CARD      00702000 D
01C5  1000 0000 0030 0000          1 = SAR                                                           00703000 D
01C6  4809 0002 0030 00F0          A2 EQL 1                % IS COL. 2 A 1?                          00704000 D
01C7  6019 1801 0030 00F0          IF TRUE THEN B101 C = A1; SKIP                                    00705000 D
01C8  4809 1800 4030 00F0          B100 = A1               % PRINTER MASK                           00706000 D
01C9  0460 0000 0030 C060          INPUT - 1 = CPCR        % SR#2 INTO B                            00707000 D
01CA  4809 AC50 0030 00F0          A1 OR B = MIR           % SET MSB OF SR # 2                       00708000 D
01CB  05E0 0000 0000 0040          OUT - 1 = MPCR                                                    00709000 D
                                                           %                                         00710000 D
                                                           %                                         00711000 D
                            % * * * * * * * * * * * * * AN/UYK-20 EMULATOR * * * * * * * * * * * * *00712000 D
                                                           %                                         00713000 D
                                                           %                                         00714000 D
                            % * * * * * * * * * * * * * UTILITY ROUTINES * * * * * * * * * * * * *00715000 D
                                                           %                                         00716000 D
                                                           %                                         00717000 D
```

```
                                        AEQM:                   % THIS ROUTINE TESTS IF THE :A: FIELD     00719000 D
                                                                % EQUALS THE :M: FIELD.  IF EQUAL LC2 SET00720000 D
                                                                % A2 = RM; B = RA RETURNED                00721000 D
01CC  4809 0000 0000 00F0               IF LC2                                                           00722000 D
01CD  4809 E000 8B0C 00F0               A3 R = B                                                         00723000 D
01CE  90F0 0000 0000 0080               4 = SAR; 15 = LIT                                                00724000 D
01CF  4809 2056 0B0C 00F0               LIT AND B = B           % B CONTAINS "A" FIELD                   00725000 D
01D0  4809 E156 2000 00F0               A3 AND LIT = A2         % A2 CONTAINS "M" FIELD                  00726000 D
01D1  4809 C052 0000 00F0               A2 EQL B                                                         00727000 D
01D2  5A49 0000 0000 00F0               IF TRUE THEN SET LC2                                             00728000 D
01D3  982D 0000 0000 00F0               JUMP                                                             00729000 D
                                                                %                                        00730000 D
                                                                %                                        00731000 D
01D4  4809 A001 2000 00F0       BUMP:   A1 L = A2               % THIS ROUTINE WILL INCREMENT THE PAR    00732000 D
01D5  0000 0000 0000 0020               COMP 16 = SAR           % TEST FOR WRAPAROUND AND CALL OPCODE    00733000 D
01D6  4809 C000 A000 00F0               A2 R = A2               % PAR ISOLATED IN LHW OF A2              00734000 D
01D7  4809 C000 A000 00F0               A2 + 1 R = A2           % INCREMENT THE COPY OF THE PAR          00735000 D
01D8  4809 C013 0000 00F0               A2 GTR 0                % CHECK FOR WRAPAROUND                   00736000 D
01D9  740B A000 4000 00F0               IF FALSE THEN A1 + 1 = A1 ELSE SKIP  % INCREMENT PAR BY 1       00737000 D
01DA  0290 0000 0000 0040               OPCODE - 1 = MPCR       % CALL OPCODE FOR NEXT INSTRUCTION       00738000 D
01DB  4809 A000 C000 00F0               A1 R = A1               % CLEAR OLD PAR                          00739000 D
01DC  0000 0000 0000 0020               15 = SAR                                                         00740000 D
01DD  4809 A001 4000 00F0               A1 L = A1                                                        00741000 D
01DE  4809 0001 0B00 00F0               1 L = B                 % 1024 INTO B                            00742000 D
01DF  A000 0000 0000 0020               COMP 10 = SAR                                                    00743000 D
01E0  4809 AC50 4000 00F0               A1 OR B = A1            % SET PAR TO 1024                        00744000 D
01E1  02A0 0000 0000 0040               OPCODE - 1 = MPCR       % FETCH NEXT INSTRUCTION                 00745000 D
                                                                %                                        00746000 D
                                                                %                                        00747000 D
                                        BYTETEST:               % THIS INSTRUCTION TEST THE LSB OF B      00748000 D
                                                                % WHICH CONTAINS EITHER (R(M)) OR (R(X)) 00749000 D
                                                                % OR (R(M+1)) AND DIVIDES B BY TWO        00750000 D
                                                                % B CONTAINS RESULT                       00751000 D
                                                                % LC2 = 1 IMPLIES LS BYTE OF Y            00752000 D
                                                                % LC2 = 0 IMPLIES MS BYTE                 00753000 D
01E2  4809 0C40 0000 00F0               B                                                                00754000 D
01E3  5A59 0000 0000 00F0               IF LST THEN SET LC2; SKIP  % IF LSB OF B EQUALS 1 THEN SET LC2 00755000 D
01E4  3809 0000 0000 00F0               IF LC2                  % ELSE CLEAR LC2                         00756000 D
01E5  4809 0C40 8B00 00F0               B R = B                 % B DIVIDED BY TWO                       00757000 D
01E6  1000 0000 0000 0000               1 = SAR                                                          00758000 D
01E7  982D 0000 0000 00F0               JUMP                                                             00759000 D
                                                                %                                        00760000 D
                                        % * * * * * * * * * * * * CARRY BIT SETTING * * * * * * * * * * * 00761000 D
                                                                %                                        00762000 D
01E8  280B 0000 0000 00F0       CARRY:  IF LC1 THEN STEP ELSE SKIP  % THIS ROUTINE WILL SET THE CARRY    00763000 D
01E9  0200 0000 0000 0040               SETCARRY - 1 = MPCR     % BIT IF LC1 IS SET UPON ENTRANCE        00764000 D
01EA  02C0 0000 0000 004C               CLEARC - 1 = MPCR       % OTHERWISE, CLEAR THE CARRY BIT         00765000 D
                                                                %                                        00766000 D
                                                                %                                        00767000 D
                                        CLEARC:                 % THIS ROUTINE CLEARS THE CARRY BIT       00768000 D
                                                                % (POSITION 20) IN PSW                    00769000 D
01EB  4809 A001 C030 00F0               A1 C = A1                                                        00770000 D
01EC  8000 0000 0000 0030               28 = SAR                                                         00771000 D
01ED  4809 A0C5 4000 40F0               A1 AND B110 = A1,CSAR                                            00772000 D
01EE  4809 A001 C030 00F0               A1 C = A1                                                        00773000 D
01EF  982D 0000 0000 00F0               JUMP                                                             00774000 D
                                                                %                                        00775000 D
                                                                %                                        00776000 D
01F0  4809 A001 C030 00F0       SETCARRY:  A1 C = A1            % THIS ROUTINE SETS THE CARRY BIT IN     00777000 D
```

```
01F1  9000 0000 0030 0030          28 = SAR                 % BIT P06. 28 IN THE PSW          00778000 D
01F2  4809 A0DC 4030 40F0          A1 OR 1 = A1,CSAR                                          00779000 D
01F3  4809 A0C1 C030 00F0          A1 C = A1                                                  00780000 D
01F4  482D 0000 0030 00F0          JUMP                                                       00781000 D
                                                            %                                 00782000 D
% * * * * * * * * * * * * * END CARRY ROUTINES * * * * * * * * * * * * *00783000 D
                                                            %                                 00784000 D
                                   CLEARBUFF:               %THIS ROUTINE INSERTS SPACES INTO THE 00785000 D
                                                            %PRINT BUFFER (33 WORDS)          00786000 D
01F5  4809 0640 2030 00F0          AMPCR = A2               % SAVE RETURN                    00787000 D
01F6  4809 0000 0931 00F0          0 = BC8,LCTR                                               00788000 D
01F7  01F0 0000 0030 00A0          COMP 16 = SAR;31 = LIT  % REQUIRED FOR BC8                 00789000 D
01F8  4809 2000 001C 00F0          LIT = MAR2                                                 00790000 D
01F9  0990 0000 0030 00E0          PRINTBUFF = LIT                                            00791000 D
01FA  4809 0C40 0030 00FC          B = MIR                  %SPACES ARE WRITTEN INTO MIR      00792000 D
01FB  020C 0000 0000 0060          PBUFF:  FOUTPUT - 1 = CPCR                                 00793000 D
01FC  8400 0F46 001E 00F0          IF NOT COV THEN PMAR + 1 = MAR2,INC; STEP ELSE SKIP        00794000 D
01FD  1FA0 0000 0030 0040          PBUFF -1 = MPCR                                            00795000 D
01FE  4809 C000 0040 00F0          A2 = AMPCR               % RESTORE RETURN                  00796000 D
01FF  4809 00C0 0030 00F0          STEP                     % DELAY                           00797000 D
0200  482D 0000 0030 00F0          JUMP                                                       00798000 D
                                                            %                                 00799000 D
                                                            %                                 00800000 D
% * * * * * * * * * * * * CONDITION CODE SETTING* * * * * * * * * * * * *00801000 D
                                                            %                                 00802000 D
                                                            %                                 00803000 D
                                   SETCC:                   % THIS ROUTINE WILL SET THE APPROPRIATE 00804000 D
                                                            % CONDITION CODE IN BITS 24-25 OF THE 00805000 D
                                                            % PSW.  THIS ROUTINE MUST BE PASSED 00806000 D
                                                            % R(A) IN A2 AND R(M) OR (Y) IN B.  00807000 D
0201  4809 CC52 0030 00F0          A2 EQL B                 % CHECK IF R(A) = P(M) OR (Y)      00808000 D
0202  6419 CC4C 0030 00F0          IF FALSE THEN A2 XOR B; SKIP                               00809000 D
0203  02E0 0000 0000 0040          SET00 - 1 = MPCR         % SET 00 CONDITION CODE           00810000 D
0204  4819 00C0 0030 00F0          IF MST THEN SKIP         % UNLIKE SIGNS CASE               00811000 D
0205  02FC 0000 0030 0040          LIKE - 1 = MPCR          % LIKE SIGNS CASE                 00812000 D
0206  4809 C000 0000 00F0          A2                       % A2 TO THE ADDER                 00813000 D
0207  4808 0000 0030 00FC          IF MST THEN STEP ELSE SKIP  % MST -> A2 WAS NEGATIVE        00814000 D
0208  030C 0000 0030 C040          SET11 - 1 = MPCR         % SET 11 CONDITION CODE           00815000 D
0209  0310 00C0 C030 C040          SET01 - 1 = MPCR         % A2 POSITIVE, B NEGATIVE         00816000 D
020A  4809 CC58 0030 00F0          LIKE:   A2 GTR B         % LIKE SIGNS CASE                 00817000 D
020B  780B 0000 0030 00F0          IF TRUE THEN STEP ELSE SKIP                                00818000 D
020C  0320 00C0 0030 0040          SET01 - 1 = MPCR         % A2 > B                          00819000 D
020D  0330 00C0 0030 0040          SET11 - 1 = MPCR         % A2 < B                          00820000 D
                                                            %                                 00821000 D
                                                            %                                 00822000 D
                                   SETCCA:                  % THIS ROUTINE TESTS THE CONTENTS OF THE 00823000 D
                                                            % DATA IN E WHICH IS THE RESULT OF AN 00824000 D
                                                            % ARITHMETIC INSTRUCTION           00825000 D
                                                            % B CAN BE 16 OR 32 BITS LC1 SET = 32 00826000 D
020E  2819 0000 0000 00F0          IF LC1 THEN SKIP         % TEST FOR DOUBLE WORD COMPARE    00827000 D
020F  4809 CC41 0B30 00F0          B L = B                                                    00828000 D
0210  0C00 0000 0030 C020          COMP 16 = SAR            % PREPARE SHIFT AMOUNT            00829000 D
0211  4809 0C52 0030 00F0          B EQL 0                                                    00830000 D
0212  6419 CC40 0030 00F0          IF FALSE THEN B; SKIP    % SKIP IF NON-ZERO                00831000 D
0213  0340 0000 0030 0040          SET00 - 1 = MPCR                                           00832000 D
0214  4819 00C0 0030 00FC          IF MST THEN SKIP         % CHECK FOR NEGATIVE VALUE        00833000 D
0215  0350 00C0 0000 0040          SET01 - 1 = MPCR                                           00834000 D
0216  0360 C000 0000 C040          SET11 - 1 = MPCR                                           00835000 D
                                                            %                                 00836000 D
                                                            %                                 00837000 D
```

```
0217  4809 18C1 8000 00F0     SET11:   8101 C = B          % DEFAULT TO (R(A)) < (R(M)) OR (Y)      00838C00 D
                                                           % ARITHMETIC TEST < 0 (NEG)             00839C00 D
0218  000C 0000 0000 001C              8 = SAR                                                     00840C00 D
0219  2809 0000 0000 00F0              IF LC1               % JUST TO CLEAR FLAG SET IN SETCC        00841000 D
021A  4809 AC5C 4000 00F0              B OR A1 = A1                                                 00842C00 D
021B  482D 000D 0030 00F0              JUMP                                                         00843C00 D
                                                           %                                       00844C00 D
                                                           %                                       00845C00 D
                                SET00:                     % THIS ROUTINE WILL SET CONDITION CODE 000C846C00 D
                                                           % IN BITS 24-25 OF THE PSW              00847C00 D
                                                           % ARITHMETIC TEST 0                     00848C00 D
021C  4809 18D9 8B30 00FC              B010 C = B                                                   00849C00 D
021D  0000 0000 0030 001C              8 = SAR                                                      00850C00 D
021E  4809 AC55 4030 00FC              B AND A1 = A1                                                00851C00 D
021F  482D 000D 003C 00F0              JUMP                                                         00852C00 D
                                SET01:                     % THIS ROUTINE WILL SET CONDITION CODE 010C853C00 D
                                                           % IN BITS 24-25 OF THE PSW              00854C00 D
                                                           % ARITHMETIC TEST > 0 (POS)             00855C00 D
0220  4809 1800 8B3C 00F0              B100 R = B                                                   00856000 D
0221  000C 0000 0000 001C              8 = SAR                                                      00857000 D
0222  4809 AC5C 403C 00FC              B OR A1 = A1                                                 00858C00 D
0223  4809 1819 8B30 00FC              B011 C = B                                                   00859000 D
0224  3C00 0000 003C 0000              7 = SAR                                                      00860C00 D
0225  4809 AC56 403C 00F0              B AND A1 = A1                                                00861C00 D
0226  482D 000D 0030 00F0              JUMP                                                         00862C00 D
                                                           %                                       00863C00 D
                                                           %                                       00864C00 D
                                CHECKCC:                   % THIS ROUTINE WILL RETURN THE 2 COND   00865C00 D
                                                           % CODE BITS IN THE LS 2 BITS OF B       00866C00 D
0227  4809 A001 C030 40F0              A1 C = A1, CSAR      % SHIFT CC BITS TO LS 2 BIT POS.         00867C00 D
0228  3030 000D 000C 00A0              23 = SAR; 3 = LIT                                            00868000 D
0229  4809 A156 0B3C 00FC              A1 AND LIT = B       % ISOLATE CC BITS IN B                  00869000 D
022A  4809 A001 C030 00F0              A1 C = A1            % RESTORE A1                            00870C00 D
022B  482D 000D 0000 00F0              JUMP                                                         00871C00 D
                                                           %                                       00872000 D
                                % * * * * * * * * * END CONDITION CODE ROUTINES* * * * * * * * * * *00873C00 D
                                                           %                                       00874C00 D
                                CONTENTSRA:                % THIS ROUTINE GETS (R(A)) IN B.  IT     00875C00 D
                                                           % ASSUMES THE INSTRUCTION IS IN B. THE  00876C00 D
                                                           % ADDRESS OF R(A) RETURNED IN MS WORD A3 00877000 D
022C  4809 0640 200C 00F0              AMPCR = A2           % SAVE RETURN ADDRESS                    00878000 D
022D  4809 0C40 8B3C 00F0              B R = B              % INSTRUCTION IS PASSED IN B             00879C00 D
022E  80F0 000D 000C 0080              4 = SAR; 15 = LIT                                            00880C00 D
022F  4809 2C56 0B3C 00FC              LIT AND B = B        % ISOLATE THE "A" FIELD                 00881C00 D
0230  0370 000D 003C 0060              REGSTACK - 1 = CPCR                                          00882C00 D
0231  4809 00CD 003C 24F0              ASE                                                          00883C00 D
0232  4809 0F40 0B3C 00F0              BMAR = B             % PUT ADDRESS OF R(A) INTO B            00884C00 D
0233  4809 0C41 0B30 00F0              B L = B              % MOVE ADDR OF R(A) TO MS WORD          00885C00 D
0234  000C 0000 0C00 C020              COMP 16 = SAR                                                00886C00 D
0235  4809 EC5C 1000 00FC              B OR A3 = A3         % UPPER WORD OF A3 HAS ADDR OF R(A)     00887C00 D
0236  0380 000D 000C 0060              FINPUT - 1 = CPCR    % (R(A)) IS IN B                        00888C00 D
0237  4809 C0CD 003C 00F0              A2 = AMPCR           % RESTORE RETURN ADDRESS                00889C00 D
0238  482D 000D 0030 00F0              JUMP                                                         00890C00 D
                                                           %                                       00891C00 D
                                                           %                                       00892C00 D
                                CONTENTSRM:                % THIS ROUTINE GETS (R(M)) IN B.  IT     00893C00 D
                                                           % ASSUMES THE INSTRUCTION IS IN A3.     00894C00 D
0239  4809 0640 2C30 00F0              AMPCR = A2           % SAVE THE RETURN ADDRESS               00895C00 D
023A  4809 E156 0B3C 00F0              A3 AND LIT = B                                               00896C00 D
023B  00FC 000D 000C C0E0              15 = LIT                                                     00897C00 D
```

```
023C  0390 0000 0000 0060          REGSTACK - 1 = CPCR                                  00898C00 D
023D  03A0 C000 0000 0060          EINPUT - 1 = CPCR      % (R(M)) INTO B               00899000 D
023E  4809 C000 0C40 00F0          A2 = AMPCR                                           00900C00 D
023F  4809 0000 0000 00F0          STEP                                                 00901000 D
0240  482D 0000 0000 00F0          JUMP                                                 00902000 D
                                                          %                             00903000 D
                                                          %                             00904000 D
                                   DIV:                   % THIS ROUTINE WILL DIVIDE A2/B, AND  00905000 D
                                                          % THE QUOTIENT IS RETURNED IN A3, REM IN 00906000 D
                                                          % A2. ALL NUMBERS MUST BE LEFT JUSTIFIED 00907000 D
                                                          % LC1 MUST BE SET FOR REG PAIR DIVIDEND  00908000 D
0241  4809 CC4C 0000 00F0          A2 XOR B                                             00909000 D
0242  4A49 0C40 0000 C0FC          B: IF MST THEN SET LC2  % NEG QUOTIENT FLAG          00910000 D
0243  4C09 0C5E 0B00 C0F0          IF MST THEN -B = B  % COMPLEMENT B                   00911000 D
0244  4809 0C40 8000 00F0          B R = B                                              00912000 D
0245  0000 0000 0000 0020          16 = SAR                                             00913000 D
0246  4809 C0C0 0000 C0F0          A2                                                   00914000 D
0247  4C0B C0C2 2000 00F0          IF MST THEN NOT A2 = A2;STEP ELSE SKIP               00915000 D
0248  4809 C0C0 2000 C0F0          A2 + 1 = A2            % TWO:S COMPLEMENT            00916000 D
0249  2409 C0C0 A000 00F0          IF NOT LC1 THEN A2 R = A2  % RIGHT JUSTIFY FOR 16 BITS 00917000 D
024A  2809 0000 1000 00F0          0 = A3; IF LC1                                       00918000 D
024B  4809 0C52 0000 00F0          0 EQL B                                              00919000 D
024C  680B 0C00 0C0C 00F0          IF TRUE THEN STEP ELSE SKIP  % CHECK FOR ZERO DIVISOR 00920000 D
024D  D3B0 0000 0000 0040          OVERFLOW - 1 = MPCR                                  00921000 D
024E  4809 0000 0001 24F0          LCTR: ASE              % SET UP CTR FOR 30 ITERATIONS 00922000 D
024F  01E0 0000 0000 00E0          30 = LIT                                             00923000 D
0250  4809 0000 1010 00F0          0 = A3; BR2            % CLEAR TRIAL DIVIDEND        00924000 D
0251  4809 0C40 0010 00F0          B = MAR2               % DIVISOR INTO BMAR           00925000 D
0252  4809 00C0 0B00 00F0          0 = B                  % CLEAR PARTIAL QUOTIENT      00926000 D
0253  4809 CF5E 0000 00F0          A2 LSS BMAR            % QUOTIENT = 0                00927000 D
0254  740B 0000 1000 00F0          IF TRUE THEN 0 = A3; STEP ELSE SKIP                  00928000 D
0255  03C0 0000 0000 0040          ZQUO - 1 = MPCR                                      00929000 D
0256  4809 C001 2000 00F0          A2 L = A2              % SHIFT OFF MSB (ZERO)        00930000 D
0257  900C 0000 0000 0030          COMP 1 = SAR                                         00931000 D
0258  4809 C000 0000 00F0          DIV1:  A2              % BRING DIVIDEND TO THE ADDER 00932000 D
0259  4C09 E0DC 1000 00F0          IF MST THEN A3 OR 1 = A3  % BUILD TRIAL DIVIDEND     00933000 D
025A  4809 C001 2002 00F0          A2 L = A2, INC         % PREPARE NEXT MSB IN DIVIDEND 00934000 D
025B  3000 0000 0000 0030          COMP 1 = SAR                                         00935000 D
025C  4809 EF5E 0000 00F0          A3 GEQ BMAR            % CHECK IF TRIAL DIVIDEND >= DIVISOR 00936000 D
025D  780B 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP                          00937000 D
025E  D30C 00C0 0000 004C          ONET - 1 = MPCR                                      00938000 D
025F  4809 0C41 0B00 C0F0          B L = B                % PUT ZERO IN PARTIAL QUOTIENT 00939000 D
0260  900C 00C0 0000 0030          COMP 1 = SAR                                         00940000 D
0261  03E0 0000 0000 0040          DIV2 - 1 = MPCR                                      00941000 D
0262  4809 EF5E 1000 00F0          ONET:  A3 - BMAR = A3  % TRIAL DIVIDEND - DIVISOR    00942000 D
0263  4809 0000 0000 00F0          STEP                                                 00943000 D
0264  4809 0C41 0B00 00F0          B L = B                                              00944000 D
0265  3010 00C0 0000 00B0          1 = LIT; COMP 1 = SAR                                00945000 D
0266  4809 2C5C 0800 00F0          LIT OR B = B           % PUT 1 IN THE PARTIAL QUOTIENT 00946000 D
0267  4809 E001 1000 00F0          DIV2:  A3 L = A3       % PREPARE FOR NEXT DIGIT IN TRIAL DIVIDN 00947000 D
0268  3000 0000 0000 0030          COMP 1 = SAR                                         00948000 D
0269  900B 0000 0000 00F0          IF NOT COV THEN STEP ELSE SKIP                       00949000 D
026A  2570 0000 0000 0040          DIV1 - 1 = MPCR                                      00950000 D
026B  4809 E000 AC00 00F0          A3 R = A2              % REMAINDER INTO A2           00951000 D
026C  1000 0000 0000 0000          1 = SAR                                              00952000 D
026D  4809 0C40 1000 00F0          B = A3                 % QUOTIENT INTO A3            00953000 D
026E  4809 E000 8B00 00F0          A3 R = B               % EXAMINE UHW OF QUOTIENT     00954000 D
026F  0000 0000 0000 0020          16 = SAR                                             00955000 D
0270  4809 E001 1000 00F0          A3 L = A3              % CLEAR UHW OF THE QUOTIENT   00956000 D
0271  4809 E000 9000 00F0          A3 R = A3              % RIGHT JUSTIFY QUOTIENT      00957000 D
```

```
0272  4809 00C3 0010 00FC          0 = BR2
0273  4809 0C52 0030 00FC          0 EQL 0                  % CHECK FOR A 16 BIT QUOTIENT                  00959C00 D
0274  63C9 0000 003C 00F0          IF FALSE THEN SET LC1                                                   0096000C D
0275  3C0C E002 1000 00FC          IF LC2 THEN NOT A3 = A3; STEP ELSE JUMP  % LC2 = NEG QUOT.              00961C00 D
0276  4809 E0C3 1000 C0F0          A3 + 1 = A3              % COMPLEMENT QUOTIENT                          0C962C00 D
0277  4809 E0C1 1000 C0F0          A3 L = A3                % CLEAR UHW OF QUOTIENT                        0C963C00 D
0278  0000 0000 0000 0020          COMP 16 = SAR                                                           C0964C00 D
0279  4809 E0C0 9C30 00F0          A3 R = A3                % RIGHT JUSTIFY QUOTIENT                       0C965C0C D
027A  4809 C0C2 2C00 C0F0          NOT A2 = A2              % COMPLEMENT REMAINDER                        CC966C00 D
027B  4809 C0C1 2C00 00F0          A2 + 1 L = A2            % CLEAR UHW OF REMAINDER                      00967C00 D
027C  482D C000 A000 00FC          A2 R = A2; JUMP          % RIGHT JUSTIFY REMAINDER                     C0968C0C D
027D  3C0D C0C2 2C00 00F0  ZQUO:   IF LC2 THEN  NOT A2 = A2; STEP ELSE JUMP                               0C969CC0 D
027E  482D C0C0 2C00 00FC          A2 + 1 = A2; JUMP                                                       0C97C000 D
                                                            %                                               0C971000 D
                          % * * * * * * * * * * * * * * * * DEBUGGER * * * * * * * * * * * * * * * * * * 00972C00 C
                                                            %                                               00973C00 D
                          DUMPREG:                          % THIS ROUTINE WILL DUMP OUT THE               0C974C00 D
                                                            % EMULATOR:S MAIN MEMORY MAPPING THROUGH       00975000 D
                                                            % THE RTC ADDR AND THE FOLLOWING               0C976C00 D
                                                            % D-MACHINE REGISTERS:  AMPCR, A1, A2,         0C977C00 D
                                                            % A3, MIR, AND BR1                             00978C00 D
027F  4809 0000 0030 C0F0          DMI                      % SAVE MIR IN B REGISTER                       0C979C00 D
0280  4809 0640 0030 C0FC          AMPCR = MIR              % SAVE AMPCR                                   C098CC00 D
0281  4809 2003 001C C0F0          LIT = MAR2                                                              0C981C0C D
0282  0260 0003 0C0C C0E0          WORKSPACE = LIT                                                         C0982C00 D
0283  03F0 0000 0000 0060          EOUTPUT - 1 = CPCR       % WRITE AMPCR INTO WORKSPACE                  00983C00 D
0284  4809 0F45 001C C0F0          BMAR + 1 = MAR2                                                         00984C00 D
0285  4809 A0C0 005C 00FC          A1 = MIR                                                                0C985C00 D
0286  040C 0000 0000 0060          EOUTPUT - 1 = CPCR       % WRITE A1 INTO WORKSPACE + 1                 00986C00 D
0287  4809 0F45 001C C0F0          BMAR + 1 = MAR2                                                         00987C00 D
0288  4809 C000 0030 00FC          A2 = MIR                                                                0C988C00 D
0289  041C 0000 0000 0060          EOUTPUT - 1 = CPCR       % WRITE A2 INTO WORKSPACE + 2                 00989C00 D
028A  4809 0F46 001C 00F0          BMAR + 1 = MAR2                                                         0C99C000 D
028B  4809 E0C3 0C30 00F0          A3 = MIR                                                                00991000 D
028C  0420 0000 0000 0060          EOUTPUT - 1 = CPCR       % WRITE A3 INTO WORKSPACE + 3                 00992C00 D
028D  4809 0F45 001C C0FC          BMAR + 1 = MAR2                                                         00993C00 D
028E  4809 0C40 0030 00F0          B = MIR                                                                 0C994C00 D
028F  0430 0000 0000 0060          EOUTPUT - 1 = CPCR       % WRITE MIR INTO WORKSPACE + 4                0C995C00 D
0290  4809 0F45 001C C0FC          BMAR + 1 = MAR2                                                         0C996C00 D
0291  4809 0000 0000 20FC          ASR                      % REFERENCE BR1                               00997C00 D
0292  4809 0F40 8C30 C0F0          BMAR R = MIR                                                            0C998C00 D
0293  0000 0000 0030 C010          B = SAR                                                                 0C999C00 D
0294  0440 0000 0000 C060          EOUTPUT - 1 = CPCR       % WRITE BR1 INTO WORKSPACE + 5                01000000 D
0295  4809 2000 1000 C0FC          LIT = A3                 % A3 HOLDS READ ADDR                          01001C00 D
0296  000C 0000 0030 00B0          GREG1 = LIT; COMP B = SAR                                               01002C0C D
0297  4809 2001 0C20 00FC          LIT L = BR1              % USE FR1 AS HOLDER OF READ ADDRESS           01003C00 D
0298  1F40 0000 0030 C060  NEXTEIGHT:  CLEARBUFF - 1 = CPCR  % CLEAR PRINTBUFF                             01004000 D
0299  4809 E0C3 0C1C C0F0          A3 = MAR2                % SET UP READ ADDRESS                         01005C00 D
029A  4809 2000 4000 00F0          LIT = A1                 % PRINTBUFF ADDR TO BE WRITTEN INTO           01006C0C D
029B  099C 0003 0000 00E0          PRINTBUFF = LIT                                                         01007C00 D
029C  0450 0000 0000 0060  NEXTREG: EINPUT - 1 = CPCR       % READ THE CONTENTS OF A GENERAL REG          01008C0C D
                                                            % INTO THE B REGISTER                         01009C00 D
029D  4809 0F41 002C C0F0          BMAR L = BR1                                                            01010CCC D
029E  0000 0000 0000 0030          COMP B = SAR                                                            01011C0C D
029F  4809 0C41 9031 C0F0          B C = A3; LCTR           % PUT B INTO A3, INTERCHANGE MS 16 BITS       01012C00 D
02A0  0070 0003 0000 00A0          16 = SAR; 7 = LIT        % AND LS 16 BITS                              01013C00 D
02A1  4809 0003 0B9C 00F0          0 = MIR,B                % CLEAR MIR AND B                             01014C00 D
02A2  2809 E156 2000 00F0  NEXTDIGIT:  A3 AND LIT = A2; IF LC1  % A2 CONTAINS LS 4 BITS OF A REG          01015C00 D
02A3  90F0 0000 0000 0080          15 = LIT ; 4 = SAR                                                      01016C0C D
02A4  4809 E000 900C C0F0          A3 R = A3                % SHIFT A3 RIGHT TO POSITION NEXT 4 BITS       01017C0C D
```

```
02A5  4809 C153 0030 00FC   A2 GTR LIT            % CHECK IF 4 DIGIT HEX VALUE IS A-F     01018000 D
02A6  0090 0009 0030 00E0   9 = LIT                                                        01019000 D
02A7  7C06 C140 2C00 00F0   IF TRUE THEN A2 + LIT = A2; STEP ELSE SKIP  % ADD 7 FOR BCL    01020000 D
02A8  0070 0003 0000 00E0   7 = LIT                                                        01021000 D
02A9  4809 CC41 8B00 00F0   A2 + B C = B           % PLACE 4 DIGIT VALUE INTO BC HEX WORD) 01022000 D
02AA  0000 0000 0030 0010   8 = SAR                                                        01023000 D
02AB  4809 0C40 003C 00F0   B = MIR                                                        01024000 D
02AC  4809 0B02 8B02 C0F0   NOT CTR R = B, INC     % USE B TO CHECK COUNTER VALUE          01025000 D
02AD  0090 0000 0030 00BC   24 = SAR; 4 = LIT                                              01026000 D
02AE  4809 A003 001C 00F0   A1 = MAR2              % PROPER WRITE ADDRESS                  01027000 D
02AF  4809 2C52 0030 C0F0   LIT EQL B              % TEST COUNTER FOR 4 ITERATIONS          01028000 D
02B0  6BCB 0003 0030 C0F0   IF TRUE THEN SET LC1; STEP ELSE SKIP  % TRUE MEANS HALFWORD    01029000 D
02B1  0460 0003 0000 C060   EOUTPUT - 1 = CPCR     % COMPLETED-WRITE IT INTO MEMORY         01030000 D
02B2  2FC9 A0C0 4C00 00F0   IF LC1 THEN SET LC1; A1 +  1  = A1 % INCREMENT PRINTBUFF ADDR   01031000 D
02B3  2FC9 0000 0B00 00F0   IF LC1 THEN SET LC1; 0 = B  % CLEAR B FOR NEXT HALFWORD         01032000 D
02B4  28DB 0003 003C 00F0   IF LC1 THEN STEP ELSE SKIP  % 16 BITS OF THE GIVEN REGISTER    01033000 D
02B5  2A1C 0000 0000 0C4C   NEXTDIGIT - 1 = MPCR   %HAVE BEEN CONVERTED TO BCL FORMAT      01034000 D
02B6  4809 0000 CD30 00F0   BMI                    % RESTORE PARTIAL B CONSTRUCT           01035000 D
02B7  8000 0000 0030 C0F0   IF NOT COV THEN STEP ELSE SKIP                                 01036000 D
02B8  2A1C 0000 0000 0040   NEXTDIGIT - 1 = MPCR   % LOOP DONE 8 TIMES FOR 32 BITS OF EACH 01037000 D
                                                   % GENERAL REGISTER                      01038000 D
02B9  4809 A003 001C 00F0   A1 = MAR2              % PREPARE TO WRITE OUT LAST HALFWORD     01039000 D
02BA  0470 0003 0030 C060   EOUTPUT - 1 = CPCR     % AT THIS POINT, A FULL REG IS DONE      01040000 D
02BB  4809 0000 0030 20FC   ASR                    % REFERENCE BR1                          01041000 D
02BC  4809 0F40 9000 C0F0   BMAR R = A3            % RESTORE OLD READ ADDRESS IN A3         01042000 D
02BD  000C 0000 003C C010   8 = SAR                                                        01043000 E
02BE  4809 E003 1000 C0F0   A3 + 1 = A3            % SET UP NEXT READ ADDRESS               01044000 D
02BF  4809 E152 0030 C0F0   A3 EQL LIT                                                     01045000 D
02C0  008C 0003 0030 00EC   8 = LIT                                                        01046000 D
02C1  6BCB E152 0030 00F0   A3 EQL LIT; IF TRUE THEN SET LC1;STEP ELSE SKIP                01047000 D
02C2  0480 C000 003C 0060   WRITEBUFF - 1 = CPCR                                           01048000 D
02C3  01C0 0003 003C C0E0   16 = LIT                                                       01049000 D
02C4  6BCB E152 0030 C0F0   A3 EQL LIT; IF TRUE THEN SET LC1;STEP ELSE SKIP                01050000 D
02C5  0490 0000 0000 006C   WRITEBUFF - 1 = CPCR                                           01051000 D
02C6  0180 0003 0000 C0E0   24 = LIT                                                       01052000 D
02C7  6BCB E152 0000 C0F0   A3 EQL LIT; IF TRUE THEN SET LC1;STEP ELSE SKIP                01053000 D
02C8  04A0 0003 0030 0060   WRITEBUFF - 1 = CPCR                                           01054000 D
02C9  020C 0003 003C 00E0   32 = LIT                                                       01055000 D
02CA  6BCB E152 0000 C0F0   A3 EQL LIT; IF TRUE THEN STEP; SET LC1 ELSE SKIP               01056000 D
02CB  04B0 0003 003C C060   WRITEBUFF - 1 = CPCR                                           01057000 D
02CC  028C 0003 003C C0E0   40 = LIT                                                       01058000 D
02CD  6BCB E152 0030 C0F0   A3 EQL LIT; IF TRUE THEN STEP; SET LC1 ELSE SKIP               01059000 D
02CE  04C0 0003 0030 006C   WRITEBUFF - 1 = CPCR                                           01060000 D
02CF  4809 E003 001C C0F0   A3 = MAR2              % MAR2 CONTAINS ADDRESS OF NEXT REG.    01061000 D
02D0  28DB C000 0000 00F0   IF LC1 THEN STEP ELSE SKIP  % ARE ARE WE FINISHED WITH ONE LINE01062000 D
02D1  297C C003 003C C04C   NEXTEIGHT - 1 = MPCR                                           01063000 D
02D2  4809 A140 403C C0F0   A1 + LIT = A1          % PUT SPACE BETWEEN REGISTER VALUES      01064000 D
02D3  002C 0000 003C 00EC   2 = LIT                                                        01065000 D
02D4  4809 E158 0030 00F0   A3 LEQ LIT             % CHECK IF ALL REG HAVE BEEN DUMPED     01066000 D
02D5  02FC 0003 0000 00E0   47 = LIT               % CURRENTLY, ROUTINE IS SET FOR 48 REGS 01067000 D
02D6  700B 0003 0000 00F0   IF TRUE THEN STEP ELSE SKIP                                    01068000 D
02D7  29BC 0003 003C C040   NEXTREG - 1 = MPCR     % READ ANOTHER REGISTER                 01069000 D
02D8  04DC 0000 0030 0060   WRITEBUFF - 1 = CPCR   % PRINT OUT LAST 8 REGISTERS             01070000 D
02D9  1F4C 0000 003C 006C   CLEARBUFF - 1 = CPCR   % CLEAR THE PRINT BUFFER                 01071000 D
02DA  04EC 0003 0000 0060   WRITEBUFF - 1 = CPCR   % JUST TO CREATE A BLANK LINE            01072000 D
02DB  04FC 0003 0030 0060   WRITEBUFF - 1 = CPCR   % JUST TO CREATE A BLANK LINE            01073000 D
02DC  050C 0003 003C C060   WRITEBUFF - 1 = CPCR   % JUST TO CREATE A BLANK LINE            01074000 D
02DD  051C 0000 0030 0060   WRITEBUFF - 1 = CPCR   % JUST TO CREATE A BLANK LINE            01075000 D
02DE  4809 20C3 001C 00FC   LIT + 1 = MAR2         % LOAD THE ADDRESS OF A1 INTO MAR2       01076000 D
02DF  0260 0000 0030 C0E0   WORKSPACE = LIT                                                01077000 D
```

```
02E0  052C 0000 0000 0060          EINPUT - 1 = CPCR      % READ IN CONTENTS OF A1          01078000 D
02E1  4809 0C40 4000 00F0          B = A1                 % RESTORE A1                       01079000 D
02E2  4809 0F46 001C 00F0          BMAR + 1 = MAR2        % CONSTRUCT ADDRESS OF A2          01080000 D
02E3  0530 0000 0030 0060          EINPUT - 1 = CPCR      % READ IN A2                      01081000 D
02E4  4809 0C40 2030 00F0          B = A2                 % RESTORE A2                       01082000 D
02E5  4809 0F46 001C 00F0          BMAR + 1 = MAR2        % CONSTRUCT ADDRESS OF A3          01083000 D
02E6  0540 0000 0030 0060          EINPUT - 1 = CPCR      % READ IN A3                      01084000 D
02E7  4809 0C40 1030 00F0          B = A3                 % RESTORE A3                       01085000 D
02E8  4809 0F46 001C 00F0          BMAR + 1 = MAR2        % CREATE ADDR OF MIR(WORKSPACE + 4) 01086000 D
02E9  0550 0000 0030 0060          EINPUT - 1 = CPCR      % READ IN MIR                     01087000 D
02EA  4809 0C40 0030 00F0          B = MIR                % RESTORE MIR                      01088000 D
02EB  4809 0F46 001C 00F0          BMAR + 1 = MAR2        % CREATE ADDR OF BR1               01089000 D
02EC  0560 0000 0000 0060          EINPUT - 1 = CPCR                                         01090000 D
02ED  4809 0C41 0020 00F0          B L = BR1              % RESTORE BR1                      01091000 D
02EE  0260 0000 0000 00B0          COMP B = SAR ; WORKSPACE = LIT                            01092000 D
02EF  4809 2003 001C 00F0          LIT = MAR2             % PUT ADDRESS OF AMPCR IN MAR2     01093000 D
02F0  0570 0000 0030 0060          EINPUT - 1 = CPCR      % READ IN OLD AMPCR               01094000 D
02F1  4809 0C40 0040 00F0          B = AMPCR              % RESTORE OLD AMPCR                01095000 D
02F2  4809 0000 0000 00F0          STEP                                                      01096000 D
02F3  482D 0000 0000 00F0          JUMP                                                      01097000 D
                                                          %                                 01098000 D
                                                          %                                 01099000 D
                                   EINPUT:                % THIS ROUTINE READS THE CONTENT OF THE 01100000 D
                                                          % MEMORY ADDRESS IN MAR2           01101000 D
02F4  A809 0003 0030 00F0          MR2; IF RDC            % PERFORM READ                     01102000 D
02F5  AC28 00C0 0000 00F0          WHEN RDC THEN BEX; JUMP   % READ CONTENTS OF MAR2 INTO B  01103000 D
                                                          %                                 01104000 D
                                                          %                                 01105000 D
                                   EOUTPUT:               % THIS ROUTINE WRITES MIR INTO THE 01106000 D
                                                          % MEMORY ADDRESS OF MAR2           01107000 D
02F6  9809 0000 0000 1CF0          MW2; IF SAI            % MEMORY WRITE                     01108000 D
02F7  9C28 0000 0000 00F0          WHEN SAI THEN 0; JUMP  % WHEN WRITE COMPLETED JUMP       01109000 D
                                                          %                                 01110000 D
                                                          %                                 01111000 D
                                   EXTIO:                 % THIS ROUTINE DOES EXTERNAL I/O   01112000 D
                                                          % INTERFACE BETWEEN THE UYK20 EMULATION 01113000 D
                                                          % AND THE IOP.  B CONTAINS THE CONTENTS 01114000 D
                                                          % OF MAILBOX - 1  IF REQUIRED.     01115000 D
02F8  1C88 0013 0C1C 00F0          SET GC2; WHEN GC2 THEN B111=MAR2                          01116000 D
02F9  9809 0003 001C 1CF0          MW2; B110=MAR2; IF SAI                                    01117000 D
02FA  9C08 0C40 0030 00F0          WHEN SAI THEN B = MIR                                     01118000 D
02FB  4809 0013 001C 1CF0          B111 = MAR2; MW2       % WRITE B INTO MAILBOX - 1         01119000 D
02FC  9B08 0000 0000 00F0          WHEN SAI THEN SET INT                                     01120000 D
02FD  0008 0000 0030 00F0          WHEN NOT INT THEN STEP                                    01121000 D
02FE  0EC8 0000 0030 00F0          WHEN INT THEN BEX; MR2                                    01122000 D
02FF  AEC8 0C42 0030 00F0          WHEN RDC THEN NOT B; RESET GC2                            01123000 D
0300  682F 0000 0030 00F0          IF ABT THEN JUMP ELSE RETN                                01124000 D
                                                          %                                 01125000 D
                                                          %                                 01126000 D
                                   FAULT:                 % THIS ROUTINE PRINTS OUT THE ERROR 01127000 D
                                                          % MESSAGE "FAULT INTERRUPT"       01128000 D
                                                          % INSERTED BY LOADER JCL AND STOPS THE 01129000 D
                                                          % MACHINE                         01130000 D
0301  0580 0000 0000 0060          SAVEREG - 1 = CPCR                                        01131000 D
0302  4809 0641 0020 00F0          AMPCR L = BR1                                             01132000 D
0303  0CE0 0000 0000 0000          ERRORLIST = AMPCR      % CREATE ERRORLIST ADDRESS IN BR1  01133000 D
0304  0000 0000 0000 0030          COMP B = SAR                                              01134000 D
0305  059C 0000 0030 0040          PRTERR - 1 = MPCR                                         01135000 D
                                                          %                                 01136000 D
                                                          %                                 01137000 D
```

```
                                                % * * * * * * * * INPUT OUTPUT CONTROLLER * * * * * * * * * * * * * * * * * * * 01138C00 D
                                                %                                                                                 01139C00 D
                                COMPRES:        % THIS ROUTINE COMPRESSES /N/UYK20                                               01140C00 D
                                                % 16 BIT BUFFERS (LHW OF 32 BIT BURROUGHS                                        01141C00 D
                                                % S-MEMORY WORD) INTO A 32 BIT WORD                                              01142C00 D
                                                % FORMAT COMPATIBLE WITH THE IOP                                                 01143000 D
                                                % UPON ENTRY, BR1 CONTAINS BUFFER ADDRESS                                        01144C00 D
                                                % NO. OF BUFFERS IN UHW OF A3                                                    01145C00 D
                                                % THIS ROUTINE ACCEPTS A VARIABLE                                                01146C00 D
                                                % BUFFER LENGTH FOR CRT OUTPUT, 33 WORDS                                         01147C00 D
                                                % FOR FRT OUTPUT                                                                 01148C00 D
0306  4809 0640 0030 00F0                       AMPCR = MIR     % STORE AMPCR INTO STACK                                         01149C00 D
0307  4809 2000 001C 00F0                       LIT = MAR2                                                                      01150C00 D
0308  02C0 0000 0000 00E0                       STACK = LIT                                                                     01151C00 D
0309  2F50 0000 0000 C060                       EOUTPUT - 1 = CPCR  % (STACK) = AMPCR                                           01152C00 D
030A  4809 0000 0000 20F0                       ASR             % REFERENCE BR1                                                  01153C00 D
030B  4809 0F40 0010 00F0                       BMAR = BR2      % STORE BUFFER ADDRESS IN BR2                                    01154C00 D
030C  4809 E000 9000 00F0                       A3 R = A3       % CLEAR LHW OF A3 FOR COUNT                                      01155C00 D
030D  0000 0000 0000 002C                       15 = SAR                                                                        01156C00 D
030E  4809 E001 1000 00F0                       A3 L = A3                                                                       01157C00 D
030F  05AC 0000 0000 C060       COMRPT:         EMULIN - 1 = CPCR   % GET CONTENTS OF BUFFER ADDRESS WORD                       01158C00 D
0310  4809 0C41 2000 00F0                       B L = A2        % STORE CONTENTS IN UHW OF A2                                    01159C00 D
0311  0000 0000 0000 0020                       COMP 16 = SAR                                                                   01160C00 D
0312  4809 0F47 001C 00F0                       BMAR + 1 L = BR2    % GET ADJACENT WORD ADDR.                                   01161C00 D
0313  0000 0000 0000 003C                       COMP 8 = SAR                                                                    01162C00 D
0314  05BC 0000 0000 C060                       EMULIN - 1 = CPCR   % B = CONTENTS OF ADJACENT WORD                             01163C00 D
0315  4809 CC5C 0030 00F0                       A2 OR B = MIR   % COMBINED INTO ONE 32 BIT WORD                                 01164C00 D
0316  4809 0F46 0B00 00F0                       BMAR + 1 = B    % STORE NEXT WORD PAIR ADDR. IN B                               01165C00 D
0317  4809 0000 0000 20F0                       ASR             % REFERENCE BR1                                                  01166C00 D
0318  4809 0F40 801C 40F0                       BMAR R = MAR2,CSAR  % COMPRESSED WORD DESTINATION                               01167C00 D
0319  0000 00C0 0000 001C                       8 = SAR                                                                         01168C00 D
031A  4809 CC41 0020 00F0                       B L = BR1       % STORE WORD PAIR ADDR. IN B                                    01169C00 D
031B  4809 C000 0030 24F0                       ASE             % REFERENCE BR2                                                  01170C00 D
031C  4809 0F41 0010 00F0                       BMAR L = BR2    % COMPRESSED WORD DESTINATION IN BR2                            01171C00 D
031D  05C0 C000 0000 C060                       EMULOUT - 1 = CPCR                                                              01172C00 D
031E  4809 0F40 2000 00F0                       BMAR = A2       % SAVE DESTINATION ADDR. IN B                                   01173C00 D
031F  4809 2000 001C 00F0                       LIT + 1 = MAR2  % BUFFER LENGTH IN IOCW + 2                                     01174C00 D
0320  032C 0000 0000 00E0                       IOCW = LIT                                                                      01175000 D
0321  4809 0F46 0C1C 00F0                       BMAR + 1 = MAR2     % IOCW + 2 = MAR2                                           01176C00 D
0322  2F30 0000 0000 C060                       EINPUT - 1 = CPCR   % B = BUFFER LENGTH                                         01177C00 D
0323  4809 0000 0000 20F0                       ASR             % REFERENCE BR1                                                  01178C00 D
0324  4809 0F40 C010 00F0                       BMAR = BR2      % TRANSFER NEXT WORD PAIR ADDR TO BR2                           01179C00 D
0325  4809 CC01 0020 00F0                       A2 + 1 L = BR1  % NEXT DESTINATION ADDR. IN BR1                                 01180C00 D
0326  0000 0000 0000 0030                       COMP 8 = SAR                                                                    01181C00 D
0327  4809 E0C0 1000 00F0                       A3 + 1 = A3     % INCREMENT COUNTER ( < 2 16 - 1)                               01182C00 D
0328  4809 E001 2000 00F0                       A3 L = A2       % ISOLATE COUNTER IN A2                                         01183C00 D
0329  021C 0000 0000 00AC                       COMP 16 = SAR, 33 = LIT                                                         01184C00 D
032A  4809 C0C0 A000 00F0                       A2 R = A2                                                                       01185C00 D
032B  3E59 CC52 0000 00FC                       IF LC2 THEN A2 EQL B; SET LC2; SKIP                                             01186C00 D
032C  4809 C152 0000 00F0                       A2 EQL LIT                                                                      01187C00 D
032D  5819 0000 0000 00F0                       IF TRUE THEN SKIP                                                               01188C00 D
032E  30E0 0000 0000 0040                       COMRPT - 1 = MPCR                                                               01189000 D
032F  4809 2000 001C 00FC                       LIT = MAR2      % RESTORE AMPCR FROM STACK                                      01190C00 D
0330  02C0 0000 0000 00E0                       STACK = LIT                                                                     01191C00 D
0331  2F30 0000 0000 C060                       EINPUT - 1 = CPCR                                                               01192C00 D
0332  4809 0C40 0040 00F0                       B = AMPCR                                                                       01193C00 D
0333  4809 0000 0000 00FC                       STEP            % SEPARATE AMPCR ASSIGNMENTS AND JUMP                           01194C00 D
0334  482D 0000 0000 00FC                       JUMP                                                                            01195C00 D
                                                %                                                                                 01196C00 D
                                                %                                                                                 01197C00 D
```

```
                                EXPAND:                 % THIS ROUTINE WILL TAKE A BUFFER          01198000 D
                                                        % CONTAINING 32 BIT WORDS AND EXPAND EACH01199000 D
                                                        % INTO 2 - 32 BIT WORDS, PUTTING THE INF00120000 D
                                                        % IN THE LHW OF EACH 32 BIT WORD.          01201000 D
                                                        % BR1 MUST CONTAIN THE BUFFER ADDRESS      01202000 D
                                                        % AND A3 MUST CONTAIN # OF BUFFERS/COUNT   01203000 D
0335  4809 0000 0030 20F0                ASR            % REFERENCE BR1                            01204000 D
0336  4809 0F40 A030 00F0                BMAR R = A2                                              01205000 D
0337  0000 0000 0030 0010                B = SAR                                                  01206000 D
0338  4809 E001 0B30 00F0                A3 L = B       % ISOLATE THE COUNT(BUFF EXPAND FACTOR)    01207000 D
0339  0000 0000 0030 0020                COMP 16 = SAR                                            01208000 D
033A  4809 0C40 8B30 00F0                B R = B                                                  01209000 D
033B  4809 CC40 2000 00F0                A2 + B= A2     % CALCULATE FIRST OUTPUT ADDRESS           01210000 D
033C  4809 00DE 2000 00F0                A2 - 1 = A2                                              01211000 D
033D  4809 0641 0020 00F0                AMPCR L = BR1  % RETURN IN BR1                           01212000 D
033E  0000 0000 0030 0030                COMP 8 = SAR                                             01213000 D
033F  4809 C001 0010 00F0                A2 L = BR2     % BUFFER ADDR + BEF - 1 IN BR2             01214000 D
0340  050C 0000 0000 0060   EXPRPT:      EMULIN - 1 = CPCR % CONTENTS OF 1 WORD INTO E            01215000 D
0341  4809 0C41 2000 00F0                B L = A2       % WORD CONTENTS INTO UHW OF A2             01216000 D
0342  0000 0000 0000 0020                COMP 16 = SAR                                            01217000 D
0343  4809 C000 8030 24F0                A2 R = MIR: ASE                                          01218000 D
0344  4809 E001 2000 00F0                A3 L = A2      % ISOLATE BUFFER COUNT                    01219000 D
0345  4809 0000 A000 00F0                A2 R = A2                                                01220000 D
0346  4809 CF41 0010 00F0                A2 + BMAR L = BR2 % BUFFER ADDR + COUNT INTO BR2          01221000 D
0347  0000 0000 0030 0030                COMP 8 = SAR                                             01222000 D
0348  05EC 0000 0000 0060                EMULOUT - 1 = CPCR % OUTPUT 1 EXPANDED HALFWORD          01223000 D
0349  4809 0000 0000 24F0                ASE            % REFERENCE MAR2                          01224000 D
034A  4809 0F40 2030 00F0                BMAR = A2                                                01225000 D
034B  4809 E001 0B00 00F0                A3 L = B       % ISOLATE THE COUNT                        01226000 D
034C  0000 0000 0030 0020                COMP 16 = SAR                                            01227000 D
034D  4809 0C40 8B00 00F0                B R = B        % COUNT INTO B                            01228000 D
034E  4809 CC5F 0010 00F0                A2 - B L = BR2 % NEXT READ ADDRESS INTO BR2              01229000 D
034F  0000 0000 0000 0030                COMP 8 = SAR                                             01230000 D
0350  05F0 0000 0030 0060                EMULIN - 1 = CPCR % NEXT WORD INTO B                     01231000 D
0351  4809 0C40 8030 00F0                B R = MIR                                                01232000 D
0352  0000 0000 0000 0020                16 = SAR                                                 01233000 D
0353  4809 E00F 2030 00F0                A3 - 1 L = A2  % REDUCE THE COUNT BY 1                   01234000 D
0354  4809 C000 A030 00F0                A2 R = A2                                                01235000 D
0355  4809 CF41 0010 00F0                A2 + BMAR L = BR2 % NEXT WRITE ADDRESS                   01236000 D
0356  0000 0000 0030 0030                COMP 8 = SAR                                             01237000 D
0357  060C 0000 0000 0060                EMULOUT - 1 = CPCR % WRITE OUT NEXT EXPANDED WORD        01238000 D
0358  4809 E00E 1C0C 00F0                A3 - 1 = A3    % REDUCE THE COUNT                        01239000 D
0359  4809 E001 2030 00F0                A3 L = A2                                                01240000 D
035A  0000 0000 0030 0020                COMP 16 = SAR                                            01241000 D
035B  4809 C000 AB30 00F0                A2 R = A2, B                                             01242000 D
035C  4809 C012 0000 00F0                A2 EQL 0       % CHECK IF ALL WORDS EXPANDED             01243000 D
035D  5019 0000 0000 00F0                IF FALSE THEN SKIP                                       01244000 D
035E  0610 0000 0030 0040                EXPEND - 1 = MPCR                                        01245000 D
035F  4809 0F40 2000 00F0                BMAR = A2                                                01246000 D
0360  4809 CC59 0010 00F0                A2 - B- 1 L = BR2 % NEXT READ ADDRESS                    01247000 D
0361  0000 0000 0000 0030                COMP 8 = SAR                                             01248000 D
0362  33F0 0000 0000 0040                EXPRPT - 1 = MPCR                                        01249000 D
0363  4809 0000 0030 20F0   EXPEND:      ASR            % REFERENCE MR1                           01250000 D
0364  4809 0F40 8040 00F0                BMAR R = AMPCR % RESTORE RETURN ADDRESS                  01251000 D
0365  0000 0000 0030 0010                B = SAR                                                  01252000 D
0366  4809 0000 0030 00F0                STEP                                                     01253000 D
0367  482D 0000 0000 00F0                JUMP                                                     01254000 D
                                                        %                                         01255000 D
                                                        %                                         01256000 D
                                EOUT:                   % THIS ROUTINE DUMPS THE BUFFERS           01257000 D
```

```
                                                  % INDICATED BY THE BUFFER ADDRESS IO      01258000 D
                                                  % PRINTER, CRT, OR DISK (NOT IMPLEMENTED)01259000 D
0368  3809 0000 0030 00F0          IF LC2         % (IOCW) = A2, MAR2 = IOCW                 01260000 D
0369  4809 00C0 0030 24F0          ASE            % REFERENCE BR2                            01261000 D
036A  4809 0F46 001C 00F0          BMAR + 1 = MAR2 % IOCW + 1                               01262000 D
036B  2F30 0000 0000 0060          EINPUT - 1 = CPCR % B = (IOCW+1)                         01263000 D
036C  4809 0041 0020 00F0          B L = BR1      % STORE (IOCW + 1) BUFFER ADDRESS IN BR1   01264000 D
036D  0000 0000 0030 0030          COMP 8 = SAR                                             01265000 D
036E  4809 0000 A000 00F0          A2 R = A2                                                01266000 D
036F  A000 0000 0030 0010          14 = SAR       % A2 HAS DEVICE CODE                      01267000 D
0370  4809 0640 0040 00F0          A2 + AMPCR = AMPCR                                       01268000 D
0371  062C 0000 0000 00C0          OUTDEV - 1 = AMPCR                                       01269000 D
0372  4809 0000 0030 C0F0          STEP                                                     01270000 D
0373  4824 0000 0030 00F0          EXEC                                                     01271000 D
                                                  %                                         01272000 D
0374  063C 0000 0000 0040  OUTDEV: SP00 - 1 = MPCR                                          01273000 D
0375  0640 0000 0000 0040          PRTO - 1 = MPCR                                          01274000 D
0376  065C 0000 0000 0040          NOTIMP - 1 = MPCR % DISK IO NOT IMPLEMENTED              01275000 D
                                                  %                                         01276000 D
                           SP00:                  % WRITE SPO FUNCTION                      01277000 D
0377  4849 0000 0000 00F0          SET LC2        % LC2 INDICATES IOCW + 1 WILL BE USED     01278000 D
0378  305C 0000 0000 0060          COMPRES - 1 = CPCR % COMPRESS BUFFER TO 32 BIT I/O FORMAT 01279000 D
0379  4809 20C0 001C 00F0  BDSPO:  LIT + 1 = MAR2 % BUFFER ADDRESS TO MIR                    01280000 D
037A  0320 0000 0030 00E0          IOCW = LIT                                               01281000 D
037B  2F30 0000 0030 0060          EINPUT - 1 = CPCR                                        01282000 D
037C  4809 0040 0080 C0F0          B = MIR                                                  01283000 D
037D  4809 2001 0F00 00F0          LIT L = BBI                                              01284000 D
037E  007C 0000 0000 00A0          7 = LIT, COMP 16 = SAR                                   01285000 D
037F  4809 0040 0080 24F0          B = MIR, ASE   % MIR CONTAINS 7/BUFFER ADDR.             01286000 D
0380  4809 0F46 001C 00F0          BMAR + 1 = MAR2 % BUFFER LENGTH INTO B (MAILBOX-1)       01287000 D
0381  2F30 0000 0000 0060          EINPUT - 1 = CPCR                                        01288000 D
0382  4809 0040 2030 00F0          B = A2         % TRANSFER BUFFER LENGTH INTO A2           01289000 D
0383  066C 0000 0000 0060          IOCIO - 1 = CPCR                                         01290000 D
0384  3780 0000 0000 0040          BDSPO - 1 = MPCR                                         01291000 D
0385  4809 0000 0080 C0F0          A2 = MIR       % SAVE A2 IN MIR                          01292000 D
0386  4809 E000 A000 C0F0          A3 R = A2      % ISOLATE COUNTER                         01293000 D
0387  000C 0000 0000 0020          16 = SAR                                                 01294000 D
0388  4809 CCDE 2030 00F0          A2 - 1 = A2    % DECREMENT COUNTER                       01295000 D
0389  4809 E001 903C 00F0          A3 C = A3      % DECREMENT COUNTER IN A3                  01296000 D
038A  4809 E0DE 1000 00F0          A3 - 1 = A3                                              01297000 D
038B  4809 E001 903C C0F0          A3 C = A3                                                01298000 D
038C  4809 0012 0030 C0F0          A2 EQL 0                                                 01299000 D
038D  6019 0000 0000 00F0          IF FALSE THEN SKIP % TEST THE COUNTER FOR ZERO           01300000 D
038E  067C 0000 0000 004C          OPCODE - 1 = MPCR                                        01301000 D
038F  4809 20C0 001C 00F0          LIT + 1 = MAR2 % REGENERATE BUFFER ADDRESS IN B          01302000 D
0390  032C 0000 0000 00E0          IOCW = LIT                                               01303000 D
0391  2F30 0000 0000 0060          EINPUT - 1 = CPCR                                        01304000 D
0392  4809 0040 2030 00F0          B = A2, BMI    % B = BUFFER LENGTH, A2 = BUFFER ADDR     01305000 D
0393  4809 0041 0B30 00F0          B L = B        % MULTIPLY B BY 2                         01306000 D
0394  3000 0000 0000 0030          COMP 1 = SAR                                             01307000 D
0395  4809 0041 0020 C0F0          A2 + B L = BR1 % NEW BUFFER ADDR IN BR1                  01308000 D
0396  0000 0000 0030 0030          COMP 8 = SAR                                             01309000 D
0397  4809 0040 0080 00F0          A2 + B = MIR   % NEW BUFFER ADDRESS IN IOCW +1           01310000 D
0398  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR                                       01311000 D
0399  376C 0000 0000 0040          SP00 - 1 = MPCR                                          01312000 D
                                                  %                                         01313000 D
                           PRTO:                  % WRITE PRT FUNCTION                      01314000 D
039A  3050 0000 0000 0060          COMPRES - 1 = CPCR % COMPRESS THE BUFFER                 01315000 D
039B  4809 20C0 001C C0F0  BPRTO:  LIT + 1 = MAR2 % REGENERATE BUFFER ADDRESS               01316000 D
039C  032C 0000 0030 00E0          IOCW = LIT                                               01317000 D
```

```
039D  2F30 0000 0090 0060          EINPUT - 1 = CPCR                                01318000 D
039E  4809 0040 0090 00F0          B = MIR                  % STORE BUFFER ADDRESS IN MIR   01319000 D
039F  4809 0001 0F30 00F0          1 L = BPI                                        01320000 D
03A0  0C00 0000 000C 0020          COMP 16 = SAR                                    01321000 D
03A1  4809 0040 0090 00F0          B = MIR                                          01322000 D
03A2  0680 0000 000C 0060          IOCIO - 1 = CPCR                                 01323000 D
03A3  39A0 0000 0000 0040          PPRTO - 1 = MPCR                                 01324000 D
03A4  4809 E000 A000 00FC          A3 R = A2                % ISOLATE COUNTER        01325000 D
03A5  0000 0000 000C 002C          16 = SAR                                         01326000 D
03A6  4809 0CDE 2000 00F0          A2 - 1 = A2              % DECREMENT COUNTER      01327000 D
03A7  4809 E001 9000 00F0          A3 C = A3                % DECREMENT COUNTER IN A3 01328000 D
03A8  4809 EDDE 1000 00F0          A3 - 1 = A3                                      01329000 D
03A9  4809 E001 9000 00F0          A3 C = A3                % RESTORE A3             01330000 D
03AA  4809 C012 003C 00FC          A2 EQL 0                 % IS COUNTER = 0         01331000 D
03AB  6019 0003 0090 00F0          IF FALSE THEN SKIP                               01332000 D
03AC  0690 0000 000C 0040          OPCODE - 1 = MPCR                                01333000 D
03AD  4809 20C3 001C 00F0          LIT + 1 = MAR2           % REGENERATE BUFFER ADDRESS 01334000 D
03AE  0320 0000 000C 00E0          IOCW = LIT                                       01335000 D
03AF  2F30 0000 0000 0060          EINPUT - 1 = CPCR                                01336000 D
03B0  4809 2041 0020 00F0          LIT + B L = BR1          % WRITE NEW ADDRESS IN BR1 01337000 D
03B1  042C 0000 0000 00B0          66 = LIT; COMP 8 = SAR                           01338000 D
03B2  4809 2040 0090 00F0          LIT + B = MIR            % WRITE NEW BUFFER ADDR IN IOCW +1 01339000 D
03B3  2F50 0000 000C 0060          EOUTPUT - 1 = CPCR                               01340000 D
03B4  3990 0003 0000 0040          PRIO - 1 = MPCR                                  01341000 D
                                                            %                        01342000 D
                                IN:                         % THIS ROUTINE INPUTS DATA FROM 01343000 D
                                                            % PERIPHERAL DEVICES TO THE DESIGNATED 01344000 D
                                                            % BUFFERS, CRD READER, CRT, OR DISK (N1) 01345000 D
03B5  3809 0000 0030 00F0          IF LC2                   % CLEAR LC2               01346000 D
03B6  4809 C003 A000 00F0          A2 R = A2                % A2 = (IOCW)             01347000 D
03B7  A000 0000 0030 0010          19 = SAR                 % A2 CONTAINS DEVICE CODE 01348000 D
03B8  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                               01349000 D
03B9  06A0 0000 0030 00C0          INDEV - 1 = AMPCR                                01350000 D
03BA  4809 0000 0000 00F0          STEP                                             01351000 D
03BB  4824 0003 0030 00F0          EXEC                                             01352000 D
                                                            %                        01353000 D
03BC  06B0 0003 0030 0040       INDEV: SPOI - 1 = MPCR                              01354000 D
03BD  06C0 0000 0030 0040          CRDI - 1 = MPCR                                  01355000 D
03BE  06D0 0003 0000 0040          NOTIMP - 1 = MPCR        % DISK NOT IMPLEMENTED  01356000 D
                                                            %                        01357000 D
                                SPOI:                       % READ SPO FUNCTION      01358000 D
03BF  4809 20C3 001C 00F0          LIT + 1 = MAR2           % RETRIEVE BUFFER ADDR.  01359000 D
03C0  032C 0000 0000 00E0          IOCW = LIT                                       01360000 D
03C1  2F30 0000 0000 0060          EINPUT - 1 = CPCR                                01361000 D
03C2  4809 0C40 0090 00F0          B = MIR                  % STORE BUFFER ADDR IN MIR 01362000 D
03C3  4809 0C41 0020 00F0          6 L = BR1                % STORE ADDRESS IN BR1   01363000 D
03C4  0000 0003 000C 0030          COMP 8 = SAR                                     01364000 D
03C5  4809 2001 0F00 00F0          LIT L = BBI                                      01365000 D
03C6  0060 0000 0030 00A0          6 = LIT; COMP 16 = SAR                           01366000 D
03C7  4809 0C40 0090 00F0          B = MIR                                          01367000 D
03C8  06E0 0003 0000 0060          IOCIO - 1 = CPCR                                 01368000 D
03C9  3BE0 0000 0000 0040          SPOI - 1 = MPCR                                  01369000 D
03CA  4809 E000 9000 00FC          A3 R = A3                % ISOLATE COUNTER        01370000 D
03CB  0000 0000 0030 0020          16 = SAR                                         01371000 D
03CC  4809 EDDE 1000 00F0          A3 - 1 = A3              % DECREMENT COUNTER      01372000 D
03CD  4809 E001 1000 00F0          A3 L = A3                % RESTORE A3, LHW CLEARED 01373000 D
03CE  4809 E150 1000 00F0          A3 OR LIT = A3           % BUFFER EXPANSION FACTOR IN LHW OF A3 01374000 D
03CF  014C 0000 000C 00E0          20 = LIT                                         01375000 D
03D0  3340 0000 0000 0060          EXPAND - 1 = CPCR        % EXPAND BUFFER         01376000 D
03D1  4809 E000 A000 00FC          A3 R = A2                % ISOLATE COUNTER        01377000 D
```

```
0302  0000 0000 0000 0020          16 = SAR                                                01378000 D
0303  4809 0012 0000 00F0          A2 EQL 0                  % IS COUNTER ZERO?            01379000 D
0304  6019 0000 0000 00F0          IF FALSE THEN SKIP                                      01380000 D
0305  06F0 0000 0000 0040          OPCODE - 1 = MPCR                                       01381000 D
0306  4809 20C0 001C 00F0          LIT + 1 = MAR2            % REGENERATE BUFFER ADDR      01382000 D
0307  0320 0000 0000 00EC          IOCW = LIT                                              01383000 D
0308  2F30 0000 0000 0060          EINPUT - 1 = CPCR                                       01384000 D
0309  4809 2040 0030 00F0          LIT + B = MIR                                           01385000 D
030A  0280 0000 0000 00E0          40 = LIT                  % NEXT BUFFER ADDR IN MIR (WITH BEF)  01386000 D
030B  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                      01387000 D
030C  3BE0 0000 0000 0040          SPOI - 1 = MPCR                                         01388000 D
                                                             %                             01389000 D
                                CRDI:                        % READ CARD READER FUNCTION   01390000 D
030D  4809 20C0 001C 00F0          LIT + 1 = MAR2            % RETRIEVE BUFFER ADDR.       01391000 D
030E  0320 0000 0000 00EC          IOCW = LIT                                              01392000 D
030F  2F30 0000 0000 0060          EINPUT - 1 = CPCR                                       01393000 D
03E0  4809 0040 0030 00F0          B = MIR                   % STORE BUFFER ADDR IN MIR    01394000 D
03E1  4809 0041 0020 00F0          B L = BR1                 % STORE ADDRESS IN BR1        01395000 D
03E2  0000 0000 0000 0030          COMP B = SAR                                            01396000 D
03E3  0700 0000 0000 0060          IOCIO - 1 = CPCR                                        01397000 D
03E4  3DC0 0000 0000 0040          CRDI - 1 = MPCR                                         01398000 D
03E5  4809 E000 9000 00F0          A3 R = A3                 % ISOLATE COUNTER             01399000 D
03E6  0000 0000 0000 0020          16 = SAR                                                01400000 D
03E7  4809 E0DE 1000 00F0          A3 - 1 = A3               % DECREMENT COUNTER           01401000 D
03E8  4809 E001 1000 00F0          A3 L = A3                 % RESTORE A3, LHW CLEARED      01402000 D
03E9  4809 E15C 1030 00F0          A3 OR LIT = A3            % BUFFER EXPANSION FACTOR IN LHW OF A3  01403000 D
03EA  0140 0000 0000 00E0          20 = LIT                                                01404000 D
03EB  3340 0000 0000 0060          EXPAND - 1 = CPCR         % EXPAND BUFFER               01405000 D
03EC  4809 E0C0 A030 00F0          A3 R = A2                 % ISOLATE COUNTER             01406000 D
03ED  0000 0000 0000 0020          16 = SAR                                                01407000 D
03EE  4809 0012 0000 00F0          A2 EQL 0                  % IS COUNTER ZERO?            01408000 D
03EF  6019 0000 0000 00F0          IF FALSE THEN SKIP                                      01409000 D
03F0  0710 0000 0000 0040          OPCODE - 1 = MPCR                                       01410000 D
03F1  4809 20C0 001C 00F0          LIT + 1 = MAR2            % REGENERATE BUFFER ADDR      01411000 D
03F2  0320 0000 0000 00EC          IOCW = LIT                                              01412000 D
03F3  2F30 0000 0000 0060          EINPUT - 1 = CPCR                                       01413000 D
03F4  4809 2040 0030 00F0          LIT + B = MIR                                           01414000 D
03F5  0280 0000 0000 00EC          40 = LIT                  % NEXT BUFFER ADDR IN MIR (WITH BEF)  01415000 D
03F6  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                      01416000 D
03F7  3DC0 0000 0000 0040          CRDI - 1 = MPCR                                         01417000 D
                                                             %                             01418000 D
                                                             %                             01419000 D
                                IOCIO:                       % THIS ROUTINE DOES EXTERNAL IO  01420000 D
                                                             % INTERFACE BETWEEN THE UYK20 IOC  01421000 D
                                                             % AND THE BURROUGHS IOP.  LC2 INDICATES  01422000 D
                                                             % IF MAILBOX -1 IS TO BE PASSED  01423000 D
03F8  1C88 0018 001C 00F0          SET GC2; WHEN GC2 THEN B111=MAR2                        01424000 D
03F9  9809 C0C9 001C 1CF0          MW2; B110=MAR2; IF SAI                                  01425000 D
03FA  9C08 0040 0030 00F0          WHEN SAI THEN B = MIR                                   01426000 D
03FB  3A08 0018 001C 1CF0          B111 = MAR2; IF LC2 THEN MW2 ELSE SKIP                  01427000 D
03FC  9B18 C000 0000 00F0          WHEN SAI THEN SET INT; SKIP                             01428000 D
03FD  4909 0000 0030 00F0          SET INT                                                 01429000 D
03FE  D008 0000 0000 00F0          WHEN NOT INT THEN STEP                                  01430000 D
03FF  DE08 0000 0000 00F0          WHEN INT THEN BEX; MR2                                  01431000 D
0400  AEC8 0042 0030 00F0          WHEN ROC THEN NOT B; RESET GC2                          01432000 D
0401  682F 0000 0000 00F0          IF ABT THEN JUMP ELSE RETN                              01433000 D
                                                             %                             01434000 D
                                                             %                             01435000 D
                                % * * * * * * * * * * * * INDIRECT WORD ROUTINES * * * * * * * * * * * * *01436000 D
                                                             %                             01437000 D
```

```
                                                  % IW1 AT Y = :Y: + (R(M))                  01439000 D
                                                  % LC2 INDICATES BYTE INSTRUCTION            01440000 D
                                                  % ROUTINES RETURN B = Y                     01441000 D
0402  4809 2000 001C 00F0          LIT = MAR2                                                 01442000 D
0403  02C0 0000 0080 00E0          STACK = LIT                 % STORE A3 WITH RETURN ADDR. IN STACK 01443000 D
0404  4809 E000 0090 00F0          A3 = MIR                                                   01444000 D
0405  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR          % WRITE TO EMULATION RESERVE BUFFER 01445000 D
0406  0720 0000 0000 0060          IFETCH - 1 = CPCR                                          01446000 D
0407  4809 E001 1000 00F0          A3 L = A3                   % CLEAR UHW OF A3               01447000 D
0408  0000 0000 0030 0020          COMP 16 = SAR                                              01448000 D
0409  4809 E000 9000 00F0          A3 R = A3                                                  01449000 D
040A  4809 0C41 0B30 00F0          B L = B                     % UHW OF A3 CONTAINS:Y:         01450000 D
040B  4809 EC5C 1000 00F0          A3 OR B = A3                % LHW OF A3 CONTAINS INSTRUCTION 01451000 D
040C  2380 0000 0070 0060          CONTENTSRM-1 = CPCR         % B = (R(M))                   01452000 D
040D  4809 E000 A000 00F0          A3 R = A2                   % A2 CONTAINS :Y:               01453000 D
040E  0000 0000 0030 0020          16 = SAR                                                   01454000 D
040F  4809 CC40 0B30 00F0          A2 + B = B                  % CALCULATE ADDRESS OF Y        01455000 D
0410  0730 0000 0030 0040          INDIRWD - 1 = MPCR                                         01456000 D
                                                  %                                          01457000 D
                                                  %                                          01458000 D
                                  IAWOI:                      % INDIRECT ADDRESSING WITHOUT INDEXING 01459000 D
                                                  % IW AT Y = :Y:                             01460000 D
                                                  % LC2 INDICATES BYTE INST.; ROUTINE         01461000 D
                                                  % RETURNS B = Y AND A2 = (R(M))             01462000 D
0411  4809 2000 001C 00F0          LIT = MAR2                                                 01463000 D
0412  02C0 0000 0030 00E0          STACK = LIT                 % WRITE A3 WITH RETURN IN UHW   01464000 D
0413  4809 E000 0090 00F0          A3 = MIR                                                   01465000 D
0414  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                         01466000 D
0415  0740 0000 0000 0060          IFETCH - 1 = CPCR           % RETRIEVE :Y: FIELD            01467000 D
0416  0750 0000 0000 0040          INDIRWD - 1 = MPCR                                         01468000 D
                                                  %                                          01469000 D
                                                  %                                          01470000 D
                                  INDIRWD:                    % THIS ROUTINE READS IN IW1 AND TESTS 01471000 D
                                                  % IF CASCADED OR DIRECT ADDRESSING IS       01472000 D
                                                  % REQUIRED                                  01473000 D
0417  4809 E001 1000 00F0          A3 L = A3                   % CLEAR UHW OF A3               01474000 D
0418  0000 0000 0030 0020          COMP 16 = SAR                                              01475000 D
0419  4809 E000 9000 00F0          A3 R = A3                                                  01476000 D
041A  4809 0C41 0010 00FC          B L = BR2                                                  01477000 D
041B  0000 0000 0000 003C          COMP 8 = SAR                % BR2 = IW1 ADDRESS             01478000 D
041C  4809 0C41 0B30 00F0          B L = B                                                    01479000 D
041D  0000 0000 0000 0020          COMP 16 = SAR               % SAVE IW1 ADDR IN UHW OF A3   01480000 D
041E  4809 EC5C 1000 00FC          A3 OR B = A3                                               01481000 D
041F  0760 0000 0000 0060          EMULIN - 1 = CPCR           % B = (IW1)                    01482000 D
0420  4809 0C40 A000 00F0          B R = A2                    % TEST IF BIT 14 OF IW1 IS SET  01483000 D
0421  A000 0000 0030 0010          14 = SAR                                                   01484000 D
0422  4809 C000 0030 00F0          A2                                                         01485000 D
0423  580B 0000 0000 C0F0          IF LST THEN STEP ELSE SKIP                                 01486000 D
0424  0770 0000 0000 0040          CASCADED - 1 = MPCR         % IF BIT SET, CASCADED IW      01487000 D
0425  0780 0000 0000 0040          DIRECT - 1 = MPCR           % IF BIT NOT SET, FORM Y        01488000 D
                                                  %                                          01489000 D
                                                  %                                          01490000 D
                                  CASCADED:                   % THIS ROUTINE DECIDES WHICH ADDRESSING 01491000 D
                                                  % FORMULA TO USE TO COMPUTE THE NEW         01492000 D
                                                  % LOCATION OF THE INDIRECT WORD             01493000 D
0426  4809 18C1 A030 00F0          B101 C = A2                 % A2 IS BIT MASK FOR J FIELD OF IW 01494000 D
0427  3000 0000 0030 0020          19 = SAR                                                   01495000 D
0428  4809 CC56 A030 00FC          A2 AND B R = A2                                            01496000 D
0429  9000 0000 0000 0010          12 = SAR                                                   01497000 D
```

```
042A  4809 C640 0030 00F0          A2 + AMPCR = AMPCR     % COMPUTE FORMULA DESIRED          01498000 D
042B  079C 0000 0000 C0C0          IWADDR - 1 = AMPCR                                        01499000 D
042C  4809 0000 0030 00F0          STEP                                                      01500000 D
042D  4824 0000 0000 00F0          EXEC                                                      01501000 D
                                                          %                                  01502000 D
                                                          %                                  01503000 D
042E  07A0 0000 0000 0040  IWADDR: ADT0 - 1 = MPCR                                           01504000 D
042F  0780 0000 0030 0040          ADT1 - 1 = MPCR                                           01505000 D
0430  07CC 0000 0000 0040          ADT2 - 1 = MPCR                                           01506000 D
0431  07DC 0000 0000 C040          ADT3 - 1 = MPCR                                           01507000 D
                                                          %                                  01508000 D
                                                          %                                  01509000 D
                           ADT0:                          % NEW IW AT Y = IW2                 01510000 D
0432  4809 E000 A000 00F0          A3 R = A2              % SHIFT ADDR OF IW1 INTO A2         01511000 D
0433  000C 0000 0030 0020          16 = SAR                                                  01512000 D
0434  4809 C000 0010 00F0          A2 OR 1 L = BR2        % NEXT ADDRESS OF IW                01513000 D
0435  000C 0000 0030 003C          COMP 8 = SAR                                              01514000 D
0436  07EC 0000 0030 0060          EMULIN - 1 = CPCR      % R CONTAINS IW2; Y = IW2           01515000 D
0437  4160 0000 0000 0040          INDIRWD - 1 = MPCR     % Y = NEXT ADDR OF IW1              01516000 D
                                                          %                                  01517000 D
                                                          %                                  01518000 D
                           ADT1:                          % NEW IW AT Y = IW2 + (R(X))         01519000 D
0438  4809 2C56 0B30 C0F0          B AND LIT = B                                             01520000 D
0439  00F0 0000 0030 00E0          15 = LIT               % CREATE REGISTER R(X)              01521000 D
043A  07F0 0000 0030 0060          REGSTACK - 1 = CPCR                                       01522000 D
043B  2F3C 0000 0030 0060          EINPUT - 1 = CPCR      % B = (R(X))                       01523000 D
043C  0800 0000 0030 C040          ADYF - 1 = MPCR                                           01524000 D
                                                          %                                  01525000 D
                                                          %                                  01526000 D
                           ADT2:                          % NEW INDIRECTWORD AT Y = IW2 + (R(M))  01527000 D
043D  2380 0000 0030 0C60          CONTENTSRM - 1 = CPCR   % A3 HAS INST. IN LOWER 16 BITS    01528000 D
                                                          % B = (R(M))                       01529000 D
043E  0810 0000 0000 0040          ADYF - 1 = MPCR                                           01530000 D
                                                          %                                  01531000 D
                                                          %                                  01532000 D
                           ADT3:                          % NEW IW AT Y = IW2 + (R(M+1))       01533000 D
043F  4809 E156 0B30 00F0          A3 AND LIT = B                                            01534000 D
0440  00F0 0000 0030 00E0          15 = LIT                                                  01535000 D
0441  4809 0046 0B30 C0F0          B + 1 = B              % B = RM+1                          01536000 D
0442  0820 0000 0000 0060          REGSTACK - 1 = CPCR                                       01537000 D
0443  2F3C 0000 0030 0060          EINPUT - 1 = CPCR      % B = (R(M+1))                     01538000 D
0444  083C 0000 0030 C040          ADYF - 1 = MPCR                                           01539000 D
                                                          %                                  01540000 D
                                                          %                                  01541000 D
                           ADYF:                          % THIS ROUTINE COMPUTES Y GIVEN     01542000 D
                                                          % B = (R(X)),(R(M)), OR (R(M+1))    01543000 D
0445  4809 E000 A000 00F0          A3 R = A2              % SHIFT ADDR OF IW INTO A2          01544000 D
0446  000C 0000 0030 0020          16 = SAR                                                  01545000 D
0447  4809 C000 0010 00F0          A2 OR 1 L = BR2        % NEXT ADDR OF IW                   01546000 D
0448  000C 0000 0030 0030          COMP 8 = SAR                                              01547000 D
0449  4809 0C40 2030 00F0          B = A2                 % SAVE (R(X)),(R(M)), OR (R(M+1)) IN A2  01548000 D
044A  084C 0000 0030 0060          EMULIN - 1 = CPCR      % B = (IW2)                        01549000 D
044B  4809 CC40 0B30 00F0          A2 + B = B             % Y = (IW2) + ((R(X)),(R(M)), OR (R(M+1)))01550000 D
044C  4160 0000 0000 0040          INDIRWD - 1 = MPCR     % Y = NEXT ADDR OF IW1              01551000 D
                                                          %                                  01552000 D
                                                          %                                  01553000 D
                           DIRECT:                        % THIS ROUTINE DECIDES WHAT ADDRESSING  01554000 D
                                                          % FORMULA TO COMPUTE Y - EFF. OPERAND  01555000 D
                                                          % ADDRESS                           01556000 D
044D  4809 18C1 A030 00F0          B101 C = A2                                               01557000 D
```

```
044E  3000 0000 0000 0020          19 = SAR                  % MASK :J: FIELD OF IW1          01558000 D
044F  4809 0056 A000 00F0          A2 AND B R = A2                                           01559000 D
0450  3000 0000 0000 0010          12 = SAR                  % ISOLATE BIT VALUE IN A2       01560000 D
0451  4809 0640 0000 00F0          A2 + AMPCR = AMPCR        % COMPUTE JUMP TABLE ADDRESS     01561000 D
0452  085C 0000 0000 0000          DIRADDR - 1 = AMPCR       % FOR FORMULA DESIRED            01562000 D
0453  4809 0000 0000 00F0          STEP                                                      01563000 D
0454  4824 0000 0000 00F0          EXEC                                                      01564000 D
                                                             %                               01565000 D
                                                             %                               01566000 D
0455  0860 0000 0000 0040  DIRADDR: DI0 - 1 = MPCR                                           01567000 D
0456  0870 0000 0000 0040          DI1 - 1 = MPCR                                            01568000 D
0457  0880 0000 0000 0040          DI2 - 1 = MPCR                                            01569000 D
0458  0890 0000 0000 0040          DI3 - 1 = MPCR                                            01570000 D
                                                             %                               01571000 D
                                                             %                               01572000 D
                           DI0:                              % Y = IW2                        01573000 D
0459  4809 E000 A000 00F0          A3 R = A2                 % SHIFT ADDR OF IW1 INTO A2      01574000 D
045A  0000 0000 0000 0020          16 = SAR                                                  01575000 D
045B  4809 C000 0010 00F0          A2 OR 1 L = BR2                                           01576000 D
045C  0000 0000 0000 0030          COMP 8 = SAR                                              01577000 D
045D  08A0 0000 0000 0060          EMULIN - 1 = CPCR         % B = (IW2)                     01578000 D
045E  3809 0000 0030 00F0          IF LC2                    % CLEAR BYTE BIT(IF SET) LC2 = MS BYTE 01579000 D
045F  0880 0000 0000 0040          DIRETURN - 1 = MPCR                                       01580000 D
                                                             %                               01581000 D
                                                             %                               01582000 D
                           DI1:                              % Y = IW2 + (R(X))               01583000 D
0460  4809 2056 0B30 00F0          B AND LIT = B                                             01584000 D
0461  00F0 0000 0000 00E0          15 = LIT                  % ISOLATE RX FROM IW1            01585000 D
0462  08C0 0000 0000 0060          REGSTACK - 1 = CPCR                                       01586000 D
0463  2F30 0000 0000 0060          EINPUT - 1 = CPCR         % B = (R(X))                    01587000 D
0464  080C 0000 0000 0040          DIGN - 1 = MPCR                                           01588000 D
                                                             %                               01589000 D
                                                             %                               01590000 D
                           DI2:                              % Y = IW2 + (R(M))               01591000 D
0465  4809 E156 0B30 00F0          A3 AND LIT = B                                            01592000 D
0466  00F0 0000 0000 00E0          15 = LIT                  % MASK OFF :M: FIELD :M: = RM    01593000 D
0467  08E0 0000 0000 0060          REGSTACK - 1 = CPCR                                       01594000 D
0468  2F30 0000 0000 006C          EINPUT - 1 = CPCR         % B = (R(M))                    01595000 D
0469  08F0 0000 0000 0040          DIGN - 1 = MPCR                                           01596000 D
                                                             %                               01597000 D
                                                             %                               01598000 D
                           DI3:                              % Y = IW2 + (R(M+1))             01599000 D
046A  4809 E156 0B30 00F0          A3 AND LIT = B                                            01600000 D
046B  00F0 0000 0000 00E0          15 = LIT                  % MASK OFF RM INTO B             01601000 D
046C  4809 0046 0B00 00F0          B + 1 = B                 % B = RM + 1                     01602000 D
046D  0900 0000 0000 0060          REGSTACK - 1 = CPCR                                       01603000 D
046E  2F30 0000 0000 0060          EINPUT - 1 = CPCR         % B = (R(M+1))                  01604000 D
046F  0910 0000 0000 0040          DIGN - 1 = MPCR                                           01605000 D
                                                             %                               01606000 D
                                                             %                               01607000 D
                           DIGN:                             % THIS ROUTINE COMPUTES Y, TEST FOR BYTE 01608000 D
                                                             % INST. AND RETURNS TO CALLER OF 01609000 D
                                                             % RXMFIELD                       01610000 D
0470  4809 E000 A000 00F0          A3 R = A2                 % IW1 ADDR INTO BR2 INC. BY 1    01611000 D
0471  0000 0000 0000 0020          16 = SAR                                                  01612000 D
0472  4809 C000 0010 00F0          A2 OR 1 L = BR2                                           01613000 D
0473  0000 0000 0000 0030          COMP 8 = SAR                                              01614000 D
0474  4809 0040 2000 00F0          B = A2                    % SAVE (R(X)),(R(M)), OR(R(M+1)) IN A2 01615000 D
0475  0920 0000 0000 006C          EMULIN - 1 = CPCR         % B = (IW2)                     01616000 D
0476  4809 C000 0030 00F0          A2 = MIR                                                  01617000 D
```

```
0477  4809 0C40 2D30 C0F0          B = A2,BMI          % A2 = IW2; B =(R(M)),(R(X)),(R(P+1))   01618000 D
0478  3A96 0000 0030 00FC          IF LC2 THEN SET LC2;STEP ELSE SKIP                         01619000 D
0479  1E1C 00C0 0000 006C          BYTETEST - 1 = CPCR  % DO BYTE FUNCTION IF LC2 SET         01620000 D
047A  4809 CC40 CB30 00F0          A2 + B = B           % B = Y                                01621000 D
047B  093C 0000 0000 0040          DTRETURN - 1 = MPCR                                         01622000 D
                                                        %                                      01623000 D
                                                        %                                      01624000 D
                                   DTRETURN:            % THIS ROUTINE RETURNS TO CALLER        01625000 D
                                                        % OF RXMFIELD; ENTERS WITH B = Y        01626000 D
047C  4809 0C40 2000 00F0          B = A2               % A2 = Y                                01627000 D
047D  4809 20C3 0C1C C0F0          LIT = MAR2                                                  01628000 D
047E  02CC 0000 0030 00A0          STACK = LIT; 16 = SAR                                       01629000 D
047F  2F3C C0C0 0030 C060          EINPUT - 1 = CPCR    % READ IN CONTENTS OF  STACK           01630000 D
0480  4809 0C40 8090 00F0          B R = AMPCR          % RESTORE RETURN ADDRESS IN AMPCR       01631000 D
0481  4809 C000 0B30 00F0          A2 = B               % B = Y                                01632000 D
0482  482D 0000 0030 00F0          JUMP                 % JUMP TO RXMFIELD CALLER               01633000 D
                                                        %                                      01634000 D
                                   * * * * * * * * * END INDIRECT WORD ROUTINES * * * * * * * * * * * * * *01635000 D
                                                        %                                      01636000 D
                                   LDBLE:               % LOAD DOUBLE;                          01637000 D
                                                        % THIS ROUTINE TAKES AS INPUT AN ADDR.  01638000 D
                                                        % IN B AND STORES CONTENTS OF B AND B+1 01639000 D
                                                        % INTO RA,RA+1                          01640000 D
0483  4809 E001 100C 00F0          A3 L = A3                                                   01641000 D
0484  000C 00C0 0030 0020          COMP 16 = SAR                                               01642000 D
0485  4809 E65C 1000 C0F0          A3 OR AMPCR = A3     % SAVE RETURN IN UHW OF A3             01643000 D
0486  4809 E001 9030 C0F0          A3 C = A3                                                   01644000 D
0487  4809 0C40 2000 C0F0          B = A2               % SAVE Y OR Y* IN A2                    01645000 D
0488  4809 0C41 C010 C0F0          B L = BR2                                                   01646000 D
0489  000C 00C0 CC0C 0030          COMP 8 = SAR                                                01647000 D
048A  094C C0C0 0C00 0060          EMULIN - 1 = CPCR    % (Y) OR (Y*) INTO B                   01648000 D
048B  4809 C000 001C C0F0          A2 OR 1 L = BR2      % BR2 HAS Y* +1 OR Y+1                  01649000 D
048C  0000 00C0 0030 0030          COMP 8 = SAR                                                01650000 D
048D  4809 0C41 2000 C0F0          B L = A2             % A2 = (Y*) UHW OR (Y) UHW              01651000 D
048E  000C 00C0 0030 0620          COMP 16 = SAR                                               01652000 D
048F  095C 0000 0000 0060          EMULIN - 1 = CPCR    % B = (Y*+1) OR (Y+1)                  01653000 D
0490  4809 CC5C 2000 00F0          A2 OR B = A2         % A2 = (Y*)/(Y*+1) OR (Y)/(Y+1)         01654000 D
0491  4809 E000 B030 00FC          A3 R = B             % MASK OFF :A: FIELD                    01655000 D
0492  90F0 0000 0000 0080          Q = SAR; 15 = LIT                                           01656000 D
0493  4809 2C55 0B50 00F0          LIT AND B = B        % :A: FIELD INTO B                      01657000 D
0494  4809 C000 B030 C0F0          A2 R = MIR           % (Y) OR (Y*) INTO MIR                  01658000 D
0495  000C 00C0 0C0C 0020          16 = SAR                                                    01659000 D
0496  0960 00C0 0000 0060          REGSTACK - 1 = CPCR  % R(A) INTO MAR2                       01660000 D
0497  2F50 00C0 0000 C060          EOUTPUT - 1 = CPCR   % (Y) OR (Y*) INTO R(A)                01661000 D
0498  4809 C000 0B00 C0F0          A2 = B               % SET DOUBLEWORD ARITH COMPARE          01662000 D
0499  49C9 60C0 0000 00F0          SET LC1                                                     01663000 D
049A  2000 0000 000C 006C          SETCCA - 1 = CPCR                                           01664000 D
049B  4809 C0C1 2000 00F0          A2 L = A2                                                   01665000 D
049C  001C 0003 0000 00A0          COMP 16 = SAR;1 = LIT                                       01666000 D
049D  4809 2F5C 001C C0FC          LIT OR BMAR = MAR2   % R(A+1) INTO MAR2                     01667000 D
049E  4809 C000 B030 00F0          A2 R = MIR           % (Y*+1) INTO MIR                      01668000 D
049F  2F50 0000 00C0 0060          EOUTPUT - 1 = CPCR                                          01669000 D
04A0  4809 E000 8090 00F0          A3 R = AMPCR         % RETURN ADDRESS IN AMPCR               01670000 D
04A1  4809 0000 0030 00F0          STEP                                                        01671000 D
04A2  482D 0000 0030 00F0          JUMP                                                        01672000 D
                                                        %                                      01673000 D
                                                        %                                      01674000 D
                                   MOVE:                % MOVES THE NUMBER OF CELLS SPECIFIED   01675000 D
                                                        % BY UHW OF A3 FROM ADDR SPECIFIED BY R1 01676000 D
                                                        % INTO ADDR  LOCATED IN LHW OF A3       01677000 D
```

```
                                                                % PAGE ADDRESSING IS ALLOWED                    01678000 D
04A3  4809 C640 2000 00F0                 AMPCR = A2            % SAVE RETURN IN A2                             01679000 D
04A4  4809 0000 0000 20F0        MRPT:    ASR                                                                  01680000 D
04A5  4809 0F40 0010 00F0                 BMAR = BR2            % BR1 = BR2                                     01681000 D
04A6  0970 0000 0000 0060                 EMULIN - 1 = CPCR     % B = (BR1)                                    01682000 D
04A7  4809 0C40 0030 00F0                 B = MIR               % TRANSFER B TO MIR                             01683000 D
04A8  4809 C000 0030 00F0                 A2 = B                % STORE AMPCR IN B                              01684000 D
04A9  4809 E001 0010 00F0                 A3 L = BR2            % LHW OF A3 INTO BR2                            01685000 D
04AA  0000 0000 0000 0030                 COMP B = SAR                                                         01686000 D
04AB  4809 E000 9030 00F0                 A3 R = A3             % ZERO OUT LHW OF A3                            01687000 D
04AC  0000 0000 0000 0020                 16 = SAR                                                             01688000 D
04AD  4809 E001 1000 00F0                 A3 L = A3                                                            01689000 D
04AE  4809 EC5C 1000 00F0                 A3 OR B = A3          % STORE AMPCR IN LHW OF A3                      01690000 D
04AF  098C 0000 0000 0060                 EMULOUT - 1 = CPCR                                                   01691000 D
04B0  4809 E001 2000 00F0                 A3 L = A2             % STORE AMPCR IN UHW OF A2                      01692000 D
04B1  000C 0000 0000 0020                 COMP 16 = SAR                                                        01693000 D
04B2  4809 C000 A000 00F0                 A2 R = A2             % RETURN AMPCR TO LHW OF A2                     01694000 D
04B3  4809 E000 9030 00F0                 A3 R = A3             % ZERO LHW OF A3                               01695000 D
04B4  4809 E001 1000 00F0                 A3 L = A3                                                            01696000 D
04B5  4809 0F46 0B30 00F0                 BMAR + 1 = B          % INCREMENT ADDRESS DESTINATION                 01697000 D
04B6  4809 EC5C 1030 20F0                 A3 OR B = A3; ASR     % STORE DESTINATION ADDR IN LHW OF A3           01698000 D
04B7  4809 0F40 8030 40F0                 BMAR R = B,CSAR                                                      01699000 D
04B8  000C 000D 0030 0010                 B = SAR                                                              01700000 D
04B9  4809 0C47 0020 00F0                 B + 1 L = BR1         % INCREMENT ORIGIN                              01701000 D
04BA  4809 E001 9030 00F0                 A3 C = A3             % SHIFT COUNT TO LHW OF A3                      01702000 D
04BB  0000 0000 0000 0020                 16 = SAR                                                             01703000 D
04BC  4809 E005 1B30 00F0                 A3 - 1 = A3,B         % DECREMENT COUNTER                             01704000 D
04BD  4809 0C41 0B00 00F0                 B L = B               % ISOLATE COUNTER IN LHW OF B                   01705000 D
04BE  4809 0C40 8B00 00F0                 B R = B                                                              01706000 D
04BF  4809 E001 9030 00F0                 A3 C = A3             % RESTORE A3                                   01707000 D
04C0  4809 0C52 0030 00F0                 B EQL 0                                                              01708000 D
04C1  000C 000D 000C 0030                 COMP B = SAR                                                         01709000 D
04C2  600B 0000 0000 00F0                 IF FALSE THEN STEP ELSE SKIP                                         01710000 D
04C3  4A30 0000 0000 0040                 MRPT - 1 = MPCR                                                      01711000 D
04C4  4809 C000 0040 00F0                 A2 = AMPCR                                                           01712000 D
04C5  48C9 C000 0000 00F0                 STEP                                                                 01713000 D
04C6  482D 0000 0000 00F0                 JUMP                                                                 01714000 D
                                                                %                                              01715000 D
                                                                %                                              01716000 D
                                 MULT:                          % THIS ROUTINE WILL MULTIPLY 2 VALUES.          01717000 D
                                                                % THE OPERANDS MUST BE PASSED BY A2 AND B01718000 D
                                                                % LC1 MUST BE SET IF 32 BIT NUMBERS ARE         01719000 D
                                                                % PASSED.  PRODUCT IS RETURNED IN A3.           01720000 D
04C7  4809 0000 1000 00F0                 C = A3                % HOLDS PARTIAL PRODUCT                         01721000 D
04C8  000C 000D 0030 0020                 16 = SAR                                                             01722000 D
04C9  2809 0000 0000 00F0                 IF LC1 THEN SET LC1; SKIP                                            01723000 D
04CA  099C 000D 0000 0040                 SREG - 1 = MPCR       % DOUBLE PACK SINGLE REG VALUES                 01724000 D
04CB  4809 0C52 0030 00F0        CMUL:    C EQL B                                                              01725000 D
04CC  640D CC4C 0000 0CF0                 IF FALSE THEN A2 XOR B; STEP ELSE JUMP  % CHECK FOR NEG PROD           01726000 D
04CD  4A49 C000 0000 C0F0                 A2; IF MST THEN SET LC2 % FLAG FOR NEG PRODUCT                         01727000 D
04CE  4C0B C002 2000 C0F0                 IF MST THEN NOT A2 = A2; STEP ELSE SKIP                                01728000 D
04CF  48C9 C000 2000 C0F0                 A2 + 1 = A2           % TWO:S COMPLEMENT                              01729000 D
04D0  4809 0C40 0030 00F0                 B                     % BRING B TO THE ADDER                          01730000 D
04D1  4C09 0C5E 0B30 00F0                 IF MST THEN 0 - B = B  % COMPLEMENT B                                01731000 D
04D2  2819 C000 0000 00F0                 IF LC1 THEN SKIP                                                     01732000 D
04D3  09AC 000D 0000 0040                 RREG - 1 = MPCR       % RESTORE SINGLE REG TO LS WORDS                01733000 D
04D4  48C9 C000 0000 C0F0        MULT1:   A2                    % BRING A2 TO THE ADDER                         01734000 D
04D5  5C09 EC40 1000 00F0                 IF LST THEN A3 + B = A3                                              01735000 D
04D6  4809 C000 A000 40F0                 A2 R = A2, CSAR       % SHIFT OFF LS BIT OF A2                        01736000 D
04D7  1000 0000 0000 0000                 1 = SAR                                                              01737000 D
```

```
04D8  4809 0C41 0B30 00F0          D L = B                  % MULT MULTIPLIER BY 2                  01738C00 D
04D9  4809 C000 0030 00FC          A2                       % CHECK TO SEE IF ENTIRE MULTIPLIER     01739C00 D
04DA  4809 C012 0030 00F0          A2 EOL 0                 % HAS BEEN READ                         01740C00 D
04DB  6819 0000 0000 00F0          IF TRUE THEN SKIP        % IF MULTIPLICAND = 0, THEN DONE        01741C00 D
04DC  403C 0000 0030 0040          MULT1 - 1 = MPCR                                                 01742C00 D
04DD  3C0D E002 1000 00FC          IF LC2 THEN NOT A3 = A3;STEP ELSE JUMP                           01743C00 D
04DE  4809 E0C3 1000 00F0          A3 + 1 = A3                                                      01744C00 D
04DF  4820 0003 0030 00F0          JUMP                                                             01745C00 D
04E0  4809 0C41 0B30 00F0  SREG:   B L = B                  % MOVE B TO UHW                         01746C00 D
04E1  4809 C0C1 2030 00F0          A2 L = A2                % MOVE A2 TO UHW, 16 = SAR               01747C00 D
04E2  4CA0 0000 0000 0040          CMUL - 1 = MPCR          % CONTINUE MULT OP                      01748C00 D
04E3  4809 C0C0 A00C 00F0  RREG:   A2 R = A2                % RESTORE REG TO LHW                     01749C00 D
04E4  4809 0C41 8B00 C0F0          B R = B                  % RESTORE REG TO LHW                     01750C00 D
04E5  403C 0000 0000 0040          MULT1 - 1 = MPCR                                                 01751C00 D
                                                            %                                        01752C00 D
                                                            %                                        01753C00 D
                           NOTIMP:                          % THIS ROUTINE SETS UP AN ERROR MESSAGE  01754C00 D
                                                            % "NOT IMPLEMENTED" INSERTED VIA LOADER  01755C00 D
                                                            % JCL                                    01756C00 D
04E6  098C 0003 0030 0060          SAVEREG - 1 = CPCR                                               01757C00 D
04E7  4809 0640 0B30 00F0          AMPCR = B                                                        01758C00 D
04E8  0CE0 0000 0030 00C0          ERRORLIST = AMPCR                                                01759C00 C
04E9  4809 2C41 0020 00FC          LIT + D L = BR1          % CREATE ERRORLIST ADDR. IN BR1          01760C00 D
04EA  0080 0000 0030 00B0          8 = LIT; COMP 8 = SAR                                            01761C00 D
04EB  09C0 0000 0030 0040          PRTERR - 1 = MPCR                                                01762C00 D
                                                            %                                        01763C00 D
                                                            %                                        01764C00 D
                           OVERFLOW:                        % THIS ROUTINE  WILL PRINT OUT AN ERROR  01765C00 D
                                                            % DIAGNOSTIC INSERTED VIA LOADER JCL AND 01766C00 D
                                                            % ABORTS MACHINE EXECUTION.              01767C00 D
04EC  09DC 0000 0030 0060          SAVEREG - 1 = CPCR                                               01768C00 D
04ED  4809 0640 0B30 00F0          AMPCR = B                % TRANSFER ERRORLIST ADDR TO B           01769C00 D
04EE  0CE0 0000 0030 00C0          ERRORLIST = AMPCR                                                01770C00 D
04EF  4809 2C41 0C20 00F0          LIT + B L = BR1          % CREATE ERRORLIST ADDR IN BR1           01771C00 D
04F0  0100 0000 0030 00B0          16 = LIT; COMP 8 = SAR                                           01772C00 D
04F1  09E0 0000 0030 0040          PRTERR - 1 = MPCR        % PRINT THE ERROR MESSAGE AND STOP      01773C00 D
                                                            %                                        01774C00 D
                           * * * * * * * * * * * * *OVERFLOW BIT SETTING * * * * * * * * * * * * *  01775C00 D
                                                            %                                        01776C00 D
                           CHECKOV:                         % THIS ROUTINE WILL SET THE OVERFLOW BIT 01777C00 D
                                                            % IF REQUIRED, OTHERWISE CLEAR IT        01778C00 D
                                                            % A2 HOLDS "X", B HOLDS "Y", AND MIR     01779C00 D
                                                            % HOLDS THE ALG SUM (IN MS WORDS)        01780C00 D
                                                            % IF LC2 IS SET, IT IMPLIES A NEGATIVE   01781C00 D
04F2  380B 0003 0030 00F0          IF LC2 THEN STEP ELSE SKIP  % TEST FOR A SUBTRACTION OPCODE      01782C00 D
04F3  4809 0C5E 0B30 00F0          C - B = B                                                        01783C00 D
04F4  4809 C000 0030 00F0          A2                       % A2 TO THE ADDER                        01784C00 D
04F5  4BC9 0000 0000 00FC          IF MST THEN SET LC1      % IMPLIES X IS NEGATIVE                  01785C00 D
04F6  4809 CC4C 0C30 00FC          A2 XOR B                 % CHECK FOR "LIKE" SIGNS                 01786C00 D
04F7  480B 0000 0030 00F0          IF MST THEN STEP ELSE SKIP                                       01787C00 D
04F8  09FC 0003 0000 0040          CLEAROV - 1 = MPCR       % CLEAR THE OV BIT, UNLIKE SIGNS         01788C00 D
04F9  4809 0003 0030 00F0          BMI                                                              01789C00 D
04FA  4809 0C40 0030 00F0          B                                                                01790C00 D
04FB  480B 0003 0000 00F0          IF MST THEN STEP ELSE SKIP  % CHECK FOR NEG SUM                  01791C00 D
04FC  0A0C 0003 0030 0040          SNEG - 1 = MPCR          % NEG SUM, LIKE SIGNS                   01792C00 D
04FD  280B 0000 0030 00FC          IF LC1 THEN STEP ELSE SKIP                                       01793C00 D
04FE  0A1C 0003 0000 0040          SETOVBIT - 1 = MPCR      % POS SUM, LIKE SIGNS, NEG X             01794C00 D
04FF  0A2C 0003 0030 0040          CLEAROV - 1 = MPCR       % POS SUM, LIKE SIGNS, POS X             01795C00 D
0500  280B 0003 0030 00FC  SNEG:   IF LC1 THEN STEP ELSE SKIP                                       01796C00 D
0501  0A3C 0003 0030 0040          CLEAROV - 1 = MPCR       % NEG SUM, NEG X, LIKE SIGNS             01797C00 D
```

```
0502  0A40 0000 0000 0040          SETOVBIT - 1 = MPCR    % NEG SUM, LIKE SIGNS, POS X          01798000 D
                                                          %                                     01799000 D
                                                          %                                     01800000 D
0503  280B 0000 0000 00F0   OVBIT:  IF LC1 THEN STEP ELSE SKIP  % THIS ROUTINE WILL SET THE      01801000 D
0504  0A5C 0000 0000 0040           SETOVBIT - 1 = MPCR    % OVERFLOW BIT IN SR#1 IF LC1 IS SET 01802000 D
0505  0A6C 0000 0000 0040           CLEAROV - 1 = MPCR     % OTHERWISE, CLEAR THE OVERFLOW BIT  01803000 D
                                                          %                                     01804000 D
                                                          %                                     01805000 D
0506  4809 A001 C030 00F0   CLEAROV: A1 C = A1             % THIS ROUTINE WILL CLEAR THE OVERFLOW 01806000 D
0507  3000 0000 0030 0030           27 = SAR               % BIT IN SR#1                         01807000 D
0508  4809 A0CE 4000 40F0           A1 AND B110 = A1,CSAR                                        01808000 D
0509  2809 A001 C030 00F0           A1 C = A1; IF LC1      % CLEAR LC1 FROM CHECKOV              01809000 D
050A  482D 0000 0030 00F0           JUMP                                                         01810000 D
                                                          %                                     01811000 D
                                                          %                                     01812000 D
                            SETOVBIT:                     % THIS ROUTINE SETS THE OVERFLOW BIT IN 01813000 D
                                                          % POSITION 27 OF PSW                  01814000 D
050B  4809 A001 C030 00F0           A1 C = A1                                                    01815000 D
050C  3000 0000 0030 C030           27 = SAR                                                     01816000 D
050D  4809 A0CC 4000 40F0           A1 OR 1 = A1,CSAR                                            01817000 D
050E  4809 A001 C000 00F0           A1 C = A1                                                    01818000 D
050F  482D 0000 0030 00F0           JUMP                                                         01819000 D
                                                          %                                     01820000 D
                            * * * * * * * * * * * * END OF OV BIT SETTING* * * * * * * * * * * * *01821000 D
                                                          %                                     01822000 D
                            PRTERR:                       % THIS ROUTINE IS CALLED WHENEVER AN  01823000 D
                                                          % ERROR PRINTOUT IS GENERATED.        01824000 D
                                                          % B HAS THE ERRORLIST NUMBER.         01825000 D
0510  1F40 0000 0030 0060           CLEARBUFF - 1 = CPCR                                         01826000 D
0511  0A7C 0000 0030 0060           WRITEBUFF - 1 = CPCR   % WRITE A BLANK LINE                 01827000 D
0512  4809 0640 1030 00F0           AMPCR = A3             % WRITE PRINTBUFF ADDRESS INTO A3     01828000 D
0513  0990 0000 0030 00C0           PRINTBUFF = AMPCR                                            01829000 D
0514  4809 2001 0B00 00F0           LIT L = B              % LOAD COUNT IN UHW OF A3             01830000 D
0515  0080 0000 0000 00A0           B = LIT; COMP 16 = SAR % 8 WORDS WILL BE MOVED              01831000 D
0516  4809 EC5C 1030 00F0           A3 OR B = A3                                                 01832000 D
0517  4A20 0000 0030 0060           MOVE - 1 = CPCR        % MOVE ERROR DIAGNOSTIC INTO PRINTBUFF 01833000 D
0518  0A80 0000 0000 0060           WRITEBUFF - 1 = CPCR   % WRITE ERROR LIST MESSAGE            01834000 D
0519  1F40 0000 0000 0060           CLEARBUFF - 1 = CPCR                                         01835000 D
051A  0A90 0000 0030 0060           WRITEBUFF - 1 = CPCR                                         01836000 D
051B  0AAC 0000 0030 0060           RESREG - 1 = CPCR      % RESTORE LU REGISTERS                01837000 D
051C  0AB0 0000 0000 0040           STOPTIME - 1 = MPCR                                          01838000 D
                                                          %                                     01839000 D
                                                          %                                     01840000 D
                            REGSTACK:                     % THIS ROUTINE CHECKS THE STACK        01841000 D
                                                          % DESIGNATION BIT IN SR#1 TO DETERMINE 01842000 D
                                                          % WHICH STACK THE PROGRAMMER IS USING. 01843000 D
                                                          % THE B REGISTER IS ASSUMED TO HAVE THE 01844000 D
                                                          % RELATIVE REGISTER NUMBER.            01845000 D
                                                          % WRAPAROUND IS ALLOWED.               01846000 D
051D  4809 0C40 001C 00F0           B = MAR2               % STORE THE REGISTER                  01847000 D
051E  4809 A001 C030 00F0           A1 C = A1              % POSITION STACK DESIGNATOR BIT IN LS BIT01848000 D
051F  0000 0000 0030 0020           16 = SAR               % POSITION REG BIT. IN LS BIT         01849000 D
0520  4809 A000 0030 00F0           A1                     % SEND A1 TO THE ADDER                01850000 D
0521  580B 0000 0000 00F0           IF LST THEN STEP ELSE SKIP                                   01851000 D
0522  0ACC 0000 0000 C040           STACK2 - 1 = MPCR                                            01852000 D
0523  4809 2052 0000 00F0           LIT EQL B                                                    01853000 D
0524  010C 0000 0030 00E0           16 = LIT                                                     01854000 D
0525  680B 0000 0000 00F0           IF TRUE THEN STEP ELSE SKIP                                  01855000 D
0526  4809 0000 001C 00F0           0 = MAR2                                                     01856000 D
0527  482D A001 C030 00F0           A1 C = A1; JUMP        % RESTORE A1 TO NORMAL SR#1/PAR FORMAT 01857000 D
```

```
0528  4809 2040 001C 00F0   STACK2:  LIT + B = MAR2                                  01858C00 D
0529  0100 0000 0000 00E0           16 = LIT                                         01859000 D
052A  4809 2052 0000 00F0           LIT EQL B                                        01860000 D
052B  0200 0000 0000 00E0           32 = LIT                                         01861000 D
052C  6819 0000 0000 00F0           IF TRUE THEN SKIP                                01862000 D
052D  482D A001 C000 00F0           A1 C = A1 ; JUMP                                 01863000 D
052E  4809 2000 001C 00F0           LIT = MAR2                                       01864000 D
052F  0100 0000 0000 00E0           16 = LIT                                         01865000 D
0530  482D A001 C000 00F0           A1 C = A1; JUMP                                  01866000 D
                                                                  %                  01867000 D
                                                                  %                  01868000 D
                                    RESREG:                       % THIS ROUTINE RESTORES THE REGISTERS  01869000 D
                                                                  % OF THE LU IN WORKSPACE              01870000 D
0531  4809 0640 0030 00F0           AMPCR = MIR                   % SAVE RETURN IN MIR                  01871000 D
0532  4809 2000 001C 00F0           LIT + 1 = MAR2                % MAR2 = WORKSPACE + 1                01872000 D
0533  0260 0000 0000 00E0           WORKSPACE = LIT                                                    01873000 D
0534  2F30 0000 0000 006C           EINPUT - 1 = CPCR                                                  01874000 D
0535  4809 0040 4000 00F0           B = A1                        % RESTORE A1                          01875000 D
0536  4809 0F46 001C 00F0           BMAR + 1 = MAR2               % INCREMENT WORKSPACE ADDR            01876000 D
0537  2F30 0000 0000 0060           EINPUT - 1 = CPCR                                                  01877000 D
0538  4809 0040 2000 00F0           B = A2                        % RESTORE A2                          01878000 D
0539  4809 0F46 001C 00F0           BMAR + 1 = MAR2               % INCREMENT WORKSPACE ADDR            01879000 D
053A  2F30 0000 0000 0060           EINPUT - 1 = CPCR                                                  01880000 D
053B  4809 0040 1000 00F0           B = A3                        % RESTORE A3                          01881000 D
053C  4809 2000 001C 00F0           LIT = MAR2                    % SELECT WORKSPACE BASE ADDR          01882000 D
053D  2F30 0000 0000 006C           EINPUT - 1 = CPCR                                                  01883000 D
053E  4809 0040 0090 00F0           B = MIR,BMI                   % SWAP B AND MIR                      01884000 D
053F  4809 0040 0040 00F0           B = AMPCR,BMI                 % RESTORE AMPCR AND B                 01885000 D
0540  4809 0000 0000 00F0           STEP                                                               01886000 D
0541  482D 0000 0000 00F0           JUMP                                                               01887000 D
                                                                  %                  01888000 D
                                                                  %                  01889000 D
                                    RESTPSW:                      % THIS ROUTINE WILL PLACE THE CONTENTS 01890000 D
                                                                  % OF PSW INTO THE PAR, THEN RESTORE   01891000 D
                                                                  % THE PAR TO (PAR) + 2.               01892000 D
0542  4809 0640 0030 00F0           AMPCR = MIR                   % SAVE RETURN ADDR TO OPCODE IN MIR   01893000 D
0543  4809 2000 001C 00F0           LIT = MAR2                    % ADDRESS OF PSW                      01894000 D
0544  0200 0000 0000 00E0           PSW = LIT                                                          01895000 D
0545  2F30 0000 0000 0060           EINPUT - 1 = CPCR             % (PSW) INTO B                        01896000 D
0546  4809 0040 0090 00F0           B = MIR,BMI                   % (PSW) INTO MIR, RETURN ADDR INTO B  01897000 D
0547  4809 A001 2000 00F0           A1 L = A2                     % PAR INTO A2                         01898000 D
0548  0020 0000 0000 00A0           COMP 16 = SAR; 2 = LIT                                             01899000 D
0549  4809 C000 A030 00F0           A2 R = A2                     % RIGHT JUSTIFY (PAR)                 01900000 D
054A  4809 C141 2000 00F0           A2 + LIT L = A2               % (PAR) + 2 = A2                      01901000 D
054B  4809 CC50 1000 00F0           A2 OR B = A3                  % (PAR)+2/RETURN ADDR = A3            01902000 D
054C  4809 A000 C000 00F0           A1 R = A1                     % CLEAR PAR                           01903000 D
054D  4809 A001 4030 00F0           A1 L = A1, BMI                                                     01904000 D
054E  4809 AC50 4000 00F0           A1 OR B = A1                  % (PSW) INTO PAR                      01905000 D
054F  4809 880E 4000 00F0           A1 AND B011 = A1              % CLEAR BIASED FETCH BIT              01906000 D
0550  0A00 0000 0000 0060           IFETCH - 1 = CPCR             % INSTRUCTION AT Y INTO B             01907000 D
0551  4809 0040 0030 00F0           B = MIR                                                            01908000 D
0552  4809 A000 C000 00F0           A1 R = A1                     % CLEAR PAR                           01909000 D
0553  0000 0000 0030 0020           16 = SAR                                                           01910000 D
0554  4809 A001 4030 00F0           A1 L = A1                                                          01911000 D
0555  4809 E000 8000 00F0           A3 R = B                      % (PAR) + 2 = E                       01912000 D
0556  4809 AC50 4000 00F0           A1 OR B = A1,BMI                                                   01913000 D
0557  4809 E000 0030 00F0           A3 = AMPCR                    % RESTORE RETURN ADDR, INSTR IN B     01914000 D
0558  4809 0000 0000 00F0           STEP                                                               01915000 D
0559  483F 0000 0000 00F0           RETN                                                               01916000 D
                                                                  %                  01917000 D
```

```
                                                  %                                                  01918C00 D
RII:                                              % THIS ROUTINE COMPUTES (P)+ D = P                01919000 D
                                                  % AND JUMPS TO NEXT INSTRUCTION                   01920C00 D
055A  4809 0000 0B00 00F0          C = B                                                             01921C00 D
055B  4809 E001 1000 40F0          A3 L = A3, CSAR        % EXAMINE SIGN BIT OF "D"                   01922C00 D
055C  0000 0000 0000 C010          COMP 24 = SAR                                                     01923000 D
055D  4809 E003 0030 C0FC          A3                     % A3 TO THE ADDER                           01924C00 C
055E  4C09 0019 C800 00F0          IF MST THEN B111 L = B % IF SIGN=1, FILL UPPER BYTE WITH 1:S      01925C00 D
055F  4809 E000 9000 C0F0          A3 R = A3              % RESTORE "D" FIELD TO LS BYTE              01926C00 D
0560  4809 EC50 0B00 00F0          A3 OR B L = B          % B = SIGN/"D", IN MS WORD                  01927C00 D
0561  0000 0000 0000 0020          COMP 16 = SAR                                                     01928C00 D
0562  4809 A001 1000 00F0          A1 L = A3              % SAVE THE OLD PAR IN MS WORD OF A3         01929C00 D
0563  4809 EC40 8800 00F0          A3 + B R = B           % ALGEBRAICALLY ADD (P) + D                 01930C00 D
0564  4809 A003 C000 00F0          A1 R = A1              % CLEAR OLD PAR                             01931C00 D
0565  4809 A001 4000 00F0          A1 L = A1                                                         01932C00 D
0566  4809 AC5C 4000 00F0          A1 OR B = A1           % CREATE NEW PAR                            01933C00 D
0567  0AEC 0000 0000 0040          OPCODE - 1 = MPCR                                                 01934000 D
                                                  %                                                  01935C00 D
                                                  %                                                  01936C00 D
RM:                                               % THIS ROUTINE GETS THE CONTENTS OF R(M)          01937C00 D
                                                  % :M: INDICATED IN MAR AND RETURNS                01938C00 D
                                                  % IT IN MIR                                       01939C00 D
0568  4809 0000 0000 24F0           ASE                    % REFERENCE MAR2                            01940000 D
0569  4809 0640 2000 00F0          AMPCR = A2             % SAVE RETURN ADDRESS                       01941C00 D
056A  4809 0F40 0B00 00F0          BMAR = B                                                          01942C00 D
056B  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % SELECT REGISTER STACK TO BE USED          01943C00 D
                                                          % ADDR OF GEN REG STACK RETURNED IN MAR2   01944C00 D
056C  2F30 0000 0030 0060          EINPUT - 1 = CPCR      % GET CONTENTS OF R(M)                      01945C00 D
056D  4809 C000 0040 00F0          A2 = AMPCR             % RESTORE RETURN ADDRESS                    01946C00 D
056E  4809 0C40 0C30 00F0          B = MIR                % MIR IS USED AS TEMP STORAGE               01947C00 C
056F  4800 0000 0030 00F0          WAIT                   % FOR TEST PURPOSES                         01948C00 D
0570  482D 0000 0030 00F0          JUMP                                                              01949C00 D
                                                  %                                                  01950C00 D
                                                  %                                                  01951C00 D
% * * * * * * * * * * * * RX FORMAT ROUTINES * * * * * * * * * * * * * * *01952C00 D
                                                  %                                                  01953C00 D
                                                  %                                                  01954C00 D
RXMFIELD:                                         % THIS ROUTINE ANALYZES THE "M" FIELD OF          01955C00 D
                                                  % A TYPE RX INSTRUCTION.  UPON ENTRANCE           01956C00 D
                                                  % A3 CONTAINS THE MOST SIGNIF. WORD OF            01957C00 D
                                                  % THE DOUBLEWORD INSTRUCTION.  LC2                01958C00 D
                                                  % INDICATES BYTE INSTRUCTION.  UPON EXIT          01959C00 D
                                                  % OF DAWI, DAWCI, OR INDIRECTION                  01960C00 D
                                                  % B CONTAINS Y; LC2 PASSES MS(0),LS(1)            01961C00 D
0571  4809 E001 9000 00F0           A3 C = A3              % PREPARE MS 2 BYTES OF A3 TO HOLD RETURN01962C00 D
0572  0000 0000 0000 0020          16 = SAR               % ADDRESS                                  01963C00 D
0573  4809 E65C 1000 00F0          A3 OR AMPCR = A3       % PUT RETURN ADDR IN A3(UPPER HALF)         01964C00 D
0574  4809 E001 9000 00FC          A3 C = A3              % RESTORE A3 RETURN ADDR/INSTRUCTION        01965C00 D
0575  4809 E156 0B00 00F0          A3 AND LIT = B         % MASK OFF "M" FIELD                        01966C00 D
0576  0CF0 0000 0000 00E0          15 = LIT                                                          01967C00 D
0577  4809 0052 0000 00F0          B EQL 0                % IS M = 0?                                 01968C00 D
0578  680B 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP                                       01969C00 D
0579  0AFC 0000 0000 0040          DAWDI - 1 = MPCR       % JUMP TO DIRECT ADDRESS W/O INDEXING       01970C00 D
057A  4809 2C58 0000 00F0          B GEQ LIT              % TEST FOR M = 1XXX                        01971C00 D
057B  3080 0000 0000 C0E0          B = LIT                                                           01972C00 C
057C  780B 0000 0030 00F0          IF FALSE THEN STEP ELSE SKIP                                      01973C00 D
057D  0B00 0000 0030 0040          DAWI - 1 = MPCR        % JUMP TO DIRECT ADDR WITH INDEXING        01974C00 D
057E  4809 0C40 2030 00F0          B = A2                 % SAVE B IN A2                              01975C00 D
057F  500B 0000 0000 00F0          IF NOT LST THEN STEP ELSE SKIP                                    01976C00 D
0580  3B1C 0000 0000 0040          INDIR - 1 = MPCR       % IF M = 10,12,14,16 THEN INDIRECTION       01977C00 D
```

```
0581  0B3C 0000 0000 0040          DAWI - 1 = MPCR          % CHECKWISED M = 11,13,15,17 FLMFLDMM      01978C00  D
                                                            % DIRECT ADDRESSING WITH INDEXING (R(M))  01979000  D
                                                            %                                         01980C00  D
                                                            %                                         01981C00  D
                                DAWDI:                      % THIS ROUTINE RETURNS Y IN D              01982C00  D
                                                            % AND JUMPS TO CALLING ROUTINE            01983C00  D
                                                            % OF RXMFIELD                             01984C00  D
0582  0B30 0000 0000 0060          IFETCH - 1 = CPCR        % B HAS Y                                 01985C00  D
0583  4809 E0C0 8080 00F0          A3 R = AMPCR             % RESTORE RETURN ADDR FROM RXMFIELD       01986C00  D
0584  0000 0000 0030 0020          16 = SAR                                                           01987C00  D
0585  3809 0000 0000 00F0          IF LC2                   % CLEAR BYTE BIT, LC2=0; MSBYTE OF Y      01988C00  D
0586  482D 00C0 0030 00F0          JUMP                     % RETURN TO CALLING ROUTINE OF RXMFIELD   01989C00  D
                                                            %                                         01990C00  D
                                                            %                                         01991C00  D
                                DAWI:                       % THIS ROUTINE FORMS Y = Y + (R(M))        01992C00  D
                                                            % RETURNS Y IN B                          01993C00  D
0587  0B40 0000 0030 0060          IFETCH - 1 = CPCR        % GET NEXT ADDRESS                        01994C00  D
0588  4809 0C40 2030 00F0          B = A2                   % A2 HAS "Y"                              01995C00  D
0589  4809 E155 0B00 00FC          A3 AND LIT = B           % RETRIEVE RELATIVE  REG. NO.             01996C00  D
058A  00FC 0000 0000 00E0          15 = LIT                 % MASK                                    01997C00  D
058B  51C0 0000 0000 0060          REGSTACK - 1 = CPCR      % SELECT REG. STACK TO BE USED            01998C00  D
                                                            % COMPUTED ACTUAL LOCATION IN MAR2        01999C00  D
058C  2F3C 0000 0030 C060          EINPUT - 1 = CPCR        % B HAS (R(M))                            02000C00  D
058D  3A4B 0000 0000 C0F0          IF LC2 THEN SET LC2; STEP ELSE SKIP                               02001C00  D
058E  1E10 0000 0030 0060          BYTETEST - 1 = CPCR      % CHECK FOR BYTE INSTRUCTION              02002C00  D
058F  4809 CC40 0B00 00F0          A2 + B = B               % B HAS Y                                 02003C00  D
0590  4809 E0C0 8080 00F0          A3 R = AMPCR             % RESTORE RETURN ADDR FROM RXMFIELD       02004C00  D
0591  0000 0000 0030 0020          16 = SAR                                                           02005C00  D
0592  4809 0000 0030 00F0          STEP                                                               02006C00  D
0593  482D 0000 0030 00F0          JUMP                     % RETURN TO CALLING ROUTINE OF RXMFIELD   02007C00  D
                                                            %                                         02008C00  D
                                                            %                                         02009C00  D
                                INDIR:                      % THIS ROUTINE IS CALLED IF M=10,12,14,16  02010C00  D
0594  4809 2000 001C C0F0          LIT = MAR2               % LOAD ADDR OF SR#2 REGISTER INTO MAR2    02011C00  D
0595  0220 0000 0030 00E0          STATUS2 = LIT                                                      02012C00  D
0596  2F30 0000 0000 0060          EINPUT - 1 = CPCR        % READ SR#2 INTO B TO USE SRM ROUTINES    02013C00  D
0597  4809 C152 0030 00F0          A2 EQL LIT                                                         02014C00  D
0598  008C 0000 0000 00E0          8 = LIT                                                            02015C00  D
0599  6419 C152 0030 C0F0          IF FALSE THEN A2 EQL LIT; SKIP  % TEST IF M = 10                   02016C00  D
059A  0B50 0000 0030 0040          SRM10 - 1 = MPCR         % JUMP TO SR#2 "M" FIELD ROTUINE          02017C00  D
059B  00AC 0000 0030 00E0          10 = LIT                                                           02018C00  D
059C  6419 C152 0030 C0F0          IF FALSE THEN A2 EQL LIT; SKIP  % TEST IF M = 12                   02019C00  D
059D  0B60 0000 0030 0040          SRM12 - 1 = MPCR                                                   02020C00  D
059E  00CC 0000 0030 00E0          12 = LIT                                                           02021C00  D
059F  6019 C000 0030 00FC          IF FALSE THEN SKIP       % TEST IF M = 14                         02022C00  D
05A0  0B7C 0000 0000 0040          SRM14 - 1 = MPCR                                                   02023C00  D
05A1  0B80 0000 0000 C040          SRM16 - 1 = MPCR         % DEFAULTS TO M = 16                     02024C00  D
                                                            %                                         02025C00  D
                                                            %                                         02026C00  D
                                SRM10:                      % THIS ROUTINE WILL REMOVE THE "M" FIELD  02027C00  D
                                                            % FROM SR#2 AND CALL THE APPROPRIATE      02028C00  D
                                                            % INDIRECT ADDRESSING ROUTINE             02029C00  D
05A2  4809 18C1 A030 00F0          B101 C = A2              % ISOLATE :M: FIELD = 10                  02030C00  D
05A3  3000 0000 0030 0020          23 = SAR                                                           02031C00  D
05A4  4809 CC56 8B30 C0F0          A2 AND B R = B           % B CONTAINS STATUS REG 2                 02032C00  D
05A5  0000 0000 0000 0010          8 = SAR                                                            02033C00  D
05A6  0B90 0000 0030 0040          SRMTEST - 1 = MPCR       % JUMP TO TEST CONTENTS ROUTINE           02034C00  D
                                                            %                                         02035C00  D
                                                            %                                         02036C00  D
                                SRM12:                      % THIS ROUTINE WILL REMOVE THE :M: FIELD  02037C00  D
```

```
                                                  % INDIRECT ADDRESSING ROUTINE                          02038000 D
                                                  % FROM SR#2 AND CALL THE APPROPRIATE                   02039000 D
05A7  4809 18C1 A000 00F0          B101 C = A2    % ISOLATE :M: FIELD = 12                               02040000 D
05A8  9000 0000 0030 0020          21 = SAR                                                             02041000 D
05A9  4809 CC56 8B00 00F0          A2 AND B R = B % B CONTAINS STATUS REG 2                             02042000 D
05AA  2000 00C0 0000 0010          10= SAR        % P CONTAINS CONTENTS OF :M: FIELD                    02043000 D
05AB  0BA0 00C0 0030 0040          SRMTEST - 1 = MPCR % JUMP TO TEST CONTENTS ROUTINE                   02044000 D
                                                  %                                                      02045000 D
                                                  %                                                      02046000 D
                                   SRM14:         % THIS ROUTINE WILL REMOVE THE :M: FIELD               02047000 D
                                                  % FROM SR#2 AND CALL THE APPROPRIATE                   02048000 D
                                                  % INDIRECT ADDRESSING ROUTINE                          02049000 D
05AC  4809 18C1 A030 00F0          B101 C = A2    % ISOLATE :M: FIELD = 14                               02050000 D
05AD  3000 00C0 0030 0020          17 = SAR                                                             02051000 D
05AE  4809 CC56 8B00 00F0          A2 AND B R = B % B CONTAINS STATUS REG 2                             02052000 D
05AF  8000 0000 0030 0010          12= SAR        % B CONTAINS CONTENTS OF :M: FIELD                    02053000 D
05B0  0BB0 00C0 0030 0040          SRMTEST - 1 = MPCR % JUMP TO TEST CONTENTS ROUTINE                   02054000 D
                                                  %                                                      02055000 D
                                                  %                                                      02056000 D
                                   SRM16:         % THIS ROUTINE WILL REMOVE THE :M: FIELD               02057000 D
                                                  % FROM SR#2 AND CALL THE APPROPRIATE                   02058000 D
                                                  % INDIRECT ADDRESSING ROUTINE                          02059000 D
05B1  4809 18C1 A030 00F0          B101 C = A2    % ISOLATE :M: FIELD = 16                               02060000 D
05B2  1000 00C0 0030 0020          17 = SAR                                                             02061000 D
05B3  4809 CC56 8B30 00F0          A2 AND B R = B % B CONTAINS STATUS REG 2                             02062000 D
05B4  A000 0000 0030 0010          14= SAR        % B CONTAINS CONTENTS OF :M: FIELD                    02063000 D
05B5  0BC0 0000 0030 0040          SRMTEST - 1 = MPCR % JUMP TO TEST CONTENTS ROUTINE                   02064000 D
                                                  %                                                      02065000 D
                                                  %                                                      02066000 D
                                   SRMTEST:       %  THIS ROUTINE ANALYSES THE :M: FIELD                02067000 D
                                                  % CONTENTS M = 0,1,2,3                                 02068000 D
05B6  4809 2C5E 0030 00F0          LIT GEQ B                                                            02069000 D
05B7  0010 00C0 0030 00E0          1 = LIT                                                              02070000 D
05B8  7419 2C52 0000 00F0          IF FALSE THEN LIT EQL B; SKIP  % TEST CONTENTS OF "M"(0 OR 1)         02071000 D
05B9  586C 0000 0000 0040          DAWI - 1 = MPCR % DIRECT ADDRESSING WITH INDEXING                    02072000 D
05BA  0020 0000 0000 00E0          2 = LIT                                                              02073000 D
05BB  6819 00C0 0030 00F0          IF TRUE THEN SKIP % TEST IF "M" = 2                                  02074000 D
05BC  401C 0000 0030 0040          IAWI - 1 = MPCR % JUMP TO INDIRECT ADDR WITH INDEXING                02075000 D
05BD  4100 0000 0030 0040          IAWOI - 1 = MPCR % INDIRECT ADDR WITHOUT INDEXING                    02076000 D
                                                  %                                                      02077000 D
                                   * * * * * * * * * END RX FORMAT ROUTINES * * * * * * * * * * * * * * *02078000 D
                                                  %                                                      02079000 D
                                   SAVEREG:       % THIS ROUTINE SAVES LU(LOGIC UNIT) REG.               02080000 D
05BE  4809 0000 0030 24F0          ASE            % REFERENCE BR2/MAR                                    02081000 D
05BF  4809 0C40 0030 00F0          B = MIR        % STORE B IN MIR                                       02082000 D
05C0  4809 0640 0B00 00F0          AMPCR = B      % SAVE RETURN IN B                                     02083000 D
05C1  4809 2000 001C 00F0          LIT = MAR2                                                           02084000 D
05C2  0260 0000 003C 00E0          WORKSPACE = LIT % WRITE B INTO WORKSPACE                             02085000 D
05C3  2F50 00C0 0000 0060          FOUTPUT - 1 = CPCR                                                   02086000 D
05C4  4809 0F46 001C 00F0          BMAR + 1 = MAR2                                                      02087000 D
05C5  4809 A000 0030 00F0          A1 = MIR       % WRITE A1 INTO WORKSPACE +1                          02088000 D
05C6  2F50 00C0 0000 0060          EOUTPUT - 1 = CPCR                                                   02089000 D
05C7  4809 0F46 001C 00F0          BMAR + 1 = MAR2                                                      02090000 D
05C8  4809 C000 0030 00F0          A2 = MIR       % WRITE A2 INTO WORKSPACE + 2                         02091000 D
05C9  2F50 00C0 0000 0060          EOUTPUT - 1 = CPCR                                                   02092000 D
05CA  4809 0F46 001C 00F0          BMAR + 1 = MAR2                                                      02093000 D
05CB  4809 E000 0030 00F0          A3 = MIR       % WRITE A3 INTO WORKSPACE + 3                         02094000 D
05CC  2F50 00C0 0000 0060          EOUTPUT - 1 = CPCR                                                   02095000 D
05CD  4809 0C40 0030 00F0          B = AMPCR      % RESTORE ADDRESS                                     02096000 D
05CE  4809 00C0 0030 00F0          STEP                                                                 02097000 D
```

```
05CF  482C 0000 0000 00F0          JUMP                                                    02098000 D
                                                   %                                       02099000 D
                                                   %                                       02100000 D
                                   STARTIME:       % THIS ROUTINE STARTS THE EXTERNAL      02101000 D
                                                   % CLOCK OF THE IOP.                     02102000 D
05D0  4809 2001 0B0C 00F0          LIT L = B                                               02103000 D
05D1  0020 0000 0030 C0A0          2 = LIT; COMP 16 = SAR  % PUT FUNCTION 2 INTO B         02104000 D
05D2  4809 2C50 C030 00FC          LIT OR B = MIR          % PUT START CLOCK FUNCTION INTO MIR 02105000 D
05D3  0010 0000 0030 00E0          1 = LIT                                                 02106000 D
05D4  2F70 0000 0030 C060          EXTIO - 1 = CPCR        % CALL IOP                      02107000 D
05D5  4809 0000 0000 00FC          STEP                                                    02108000 D
05D6  0B00 0000 0030 0040          OPCODE - 1 = MPCR       % START EXECUTION                02109000 D
                                                   %                                       02110000 D
                                                   %                                       02111000 D
                                   STOPTIME:       % THIS ROUTINE SAMPLES THE EXTERNAL      02112000 D
                                                   % CLOCK OF THE IOP AND INSERTS THE TIME  02113000 D
                                                   % INTO CLOCKTIME IN S-MEMORY AND CALLS   02114000 D
                                                   % THE LOADER.                            02115000 D
05D7  4809 2001 0030 00F0          LIT L = MIR             % STORE FUNCTION IN MIR          02116000 D
05D8  002C 0000 0030 C0A0          2 = LIT; COMP 16 = SAR  % PUT FUNCTION 2 IN B           02117000 D
05D9  2F70 0000 0030 0060          EXTIO - 1 = CPCR        % CALL IOP                      02118000 D
05DA  506C 0000 0000 004C          STOPTIME - 1 = MPCR     % REREAD TIME                    02119000 D
05DB  4809 0C40 C030 C0FC          B = MIR                 % PUT CONTENTS OF MAILBOX INTO MIR 02120000 D
05DC  4809 2000 001C 00F0          LIT = MAR2                                              02121000 D
05DD  0246 0000 0030 C0E0          CLOCKTIME = LIT         % CLOCKTIME ADDRESS IN MAR2     02122000 D
05DE  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR      % WRITE CLOCK INTO CLOCKTIME     02123000 D
05DF  4809 2000 001C 00F0          LIT = MAR2                                              02124000 D
05E0  0220 0000 0000 00EC          STATUS2 = LIT                                           02125000 D
05E1  2F30 0000 0000 C060          EINPUT - 1 = CPCR       % READ IN SR #2                 02126000 D
05E2  4809 0C40 C030 00FC          B                       % SR # 2 TO THE ADDER            02127000 D
05E3  4806 0000 0000 00F0          IF MST THEN STEP ELSE SKIP  % TRACE IF TRUE             02128000 D
05E4  27E0 0000 0030 C060          DUMPREG - 1 = CPCR      % DUMP AN/UYK20 REGISTERS, ETC   02129000 D
05E5  0600 0000 0030 0040          NEWCARD - 1 = MPCR      % RETURN TO LOADER FOR ADDITIONAL INPUT 02130000 D
                                                   %                                       02131000 D
                                                   %                                       02132000 D
                                   WRITEBUFF:      % THIS ROUTINE WILL PRINTOUT THE CONTENTS02133000 D
                                                   % OF PRINTBUFF TO THE CRT OR PTR         02134000 D
                                                   % A2, MIR, AND B ARE USED IN THIS ROUTINE02135000 D
05E6  4809 0640 2030 00F0          AMPCR = A2              % SAVE AMPCR                     02136000 D
05E7  4809 2000 0030 C0F0  BDOUT:  LIT = MIR               % PUT ADDRESS OF PRINTBUFF IN MIR 02137000 D
05E8  0990 0000 0030 00E0          PRINTBUFF = LIT                                         02138000 D
05E9  4809 2000 001C C0FC          LIT = MAR2              % ADDR OF SR#2 INTO MAR2         02139000 D
05EA  B220 0000 0030 00B0          STATUS2 = LIT; COMP 1 = SAR                             02140000 D
05EB  2F30 0000 0030 0060          EINPUT - 1 = CPCR       % SR#2 INTO B                   02141000 D
05EC  4809 0C41 0B30 00FC          B L = B                 % EXAMINE 2ND MS BIT OF SR#2     02142000 D
05ED  4809 0000 0000 00F0          STEP                                                    02143000 D
05EE  007C 0000 0030 00A0          COMP 16 = SAR; 7 = LIT                                  02144000 D
05EF  4809 0C40 0030 00F0          B                       % BRING B TO THE ADDER          02145000 D
05F0  4019 2001 0F30 00F0          IF MST THEN LIT L = BBI; SKIP  % CRT FUNCTION           02146000 D
05F1  4809 0001 0F30 C0FC          1 L = BBI               % PRINTER FUNCTION              02147000 D
05F2  4809 0C40 0030 00FC          B = MIR                 % MIR CONTAINS 1/PRINTBUFF OR 7/PRINTBUFF02148000 D
05F3  4809 2000 0B0C 00F0          LIT = B                 % 33 INTO B                     02149000 D
05F4  021C 0000 0000 00E0          33 = LIT                                                02150000 D
05F5  2F70 0000 0030 0060          EXTIO - 1 = CPCR        % WRITE TO CRT/PTR 33 WORDS IN PBUFF 02151000 D
05F6  5E60 0000 0030 004C          BDOUT - 1 = MPCR        % REPEAT EXTIO, I/O NOT COMPLETE 02152000 D
05F7  4809 0000 0040 00F0          A2 = AMPCR              % RESTORE AMPCR                  02153000 D
05F8  4809 0000 0000 00F0          STEP                    % REQUIRED BETWEEN 2 TYPE II INSTRUCTIONS02154000 D
05F9  482D 0000 0000 00F0          JUMP                                                    02155000 D
                                                   %                                       02156000 D
                                                   %                                       02157000 D
```

```
                                XFCODE:                 % THIS ROUTINE WILL EXAMINE THE "F" CODE 02158000 D
05FA  4809 E000 A000 00F0               A3 R = A2       % OF AN INSTRUCTION IN B AND RETURN THE   02159000 D
05FB  0000 0000 0000 C010               B = SAR         % "F" CODE IN LS 7 BITS OF A2            02160000 D
05FC  4809 C155 2000 00F0               A2 AND LIT = A2                                          02161000 D
05FD  0030 C000 0000 00EC               3 = LIT                                                  02162000 D
05FE  482D 0000 0000 00F0               JUMP                                                     02163000 D
                                                        %                                        02164000 D
                                % * * * * * * * * * * * * FETCH ROUTINES * * * * * * * * * * * * *02165000 D
                                                        %                                        02166000 D
                                BKTCMP:                 % BKPT STATUS BIT HAS BEEN SET IN SR#1    02167000 D
                                                        % BR1 CONTAINS THE ADDRESS TO BE MATCHED.02168000 D
                                                        % BR2 CONTAINS THE DIRECT ADDRESS         02169000 D
                                                        % A1,A2,A3,AMPCR MUST BE SAVED TO BE USED02170000 D
05FF  4809 C000 0030 00F0               A2 = MIR        % SAVE A2 IN MIR                          02171000 D
0600  4809 00C0 0000 24F0               ASE             % REFERENCE BR2/MAR                       02172000 D
0601  4809 0F40 A000 00F0               GMAR R = A2     % SAVE BR2 IN A2                          02173000 D
0602  0210 C000 0030 0090               B = SAR; BKPT = LIT                                      02174000 D
0603  4809 2000 001C 00F0               LIT = MAR2      % BREAKPOINT LOCATION IN MAR2             02175000 D
0604  A809 0000 0000 00F0               MR2; IF RDC     % PERFORM READ OF BKPT                    02176000 D
0605  A008 0000 0030 00F0               WHEN RDC THEN BEX % B CONTAINS BREAKPOINT                02177000 D
0606  4809 C052 0000 00F0               A2 EQL B        % BR2 EQUAL BKPT                          02178000 D
0607  680B 0000 0030 00F0               IF TRUE THEN STEP ELSE SKIP  %IF ADDRESSES MATCH STOP EXECUTION02179000 D
0608  4800 0000 0030 00F0               WAIT                                                     02180000 D
0609  4809 C001 0010 00F0               A2 L = BR2      % RESTORE BR2                            02181000 D
060A  0000 0000 0030 0030               COMP B = SAR                                             02182000 D
060B  4809 0000 0000 00F0               GMI             % RESTORE A2 TO ORIG. CONTENTS            02183000 D
060C  280B 0040 2030 00F0               B = A2; IF LC1 THEN STEP ELSE SKIP % CALLING ROUTINE CHECK 02184000 D
060D  08F0 0000 0000 C04C               BKPTOUT - 1 = MPCR  % RETURN TO EMULOUT                  02185000 D
060E  08F0 0000 0030 C04C               BKPTIN  - 1 = MPCR  % RETURN TO EMULIN                   02186000 D
                                                        %                                        02187000 D
                                                        %                                        02188000 D
                                EMULIN:                 % THIS ROUTINE READS FROM MEMORY AFTER    02189000 D
                                                        % CHECKING THE PAGING AND BREAKPOINT BITS02190000 D
                                                        % IN SR#1.  BR2 CONTAINS THE ADDRESS TO   02191000 D
                                                        % BE READ FROM.                           02192000 D
060F  2809 A000 8B00 00F0               A1 R = B; IF LC1                                         02193000 D
0610  A000 0000 0000 0020               22 = SAR        %                                        02194000 D
0611  4809 0040 0030 00F0               B               % SEND B TO THE ADDER                     02195000 D
0612  580B 0000 0030 00F0               IF LST THEN STEP ELSE SKIP  % TEST PAGING BIT            02196000 D
0613  0000 0000 0030 C040               PAGING - 1 = MPCR   % JUMP IF PAGING BIT SET             02197000 D
0614  4809 A000 8B00 00F0               A1 R = B        %                                        02198000 D
0615  8000 0000 C030 0020               20 = SAR                                                 02199000 D
0616  4809 0040 0030 00F0               B               % SEND B TO THE ADDER                     02200000 D
0617  580B 0000 0000 00F0               IF LST THEN STEP ELSE SKIP  % TEST BREAKPOINT BIT        02201000 D
0618  5FE0 0000 0000 004C               BKTCMP - 1 = MPCR   % JUMP TO BKPT ANALYSIS ROUTINE      02202000 D
0619  4809 0000 0000 24F0               ASE             % REFERENCE MAR2                          02203000 D
061A  4809 0F40 801C 00F0               GMAR R = MAR2   % SET UP MAR2 WITH ADDRESS               02204000 D
061B  0000 0000 0030 0010               B = SAR                                                  02205000 D
061C  A809 0000 0000 00F0               MR2; BEX; IF RDC    % READ IN PRESENT ADDRESS INTO B      02206000 D
061D  A008 0000 0000 00F0               WHEN RDC THEN 0                                          02207000 D
061E  482D 0000 0030 00F0               JUMP                                                     02208000 D
                                                        %                                        02209000 D
                                                        %                                        02210000 D
                                EMULOUT:                % THIS ROUTINE WRITES OUT INTO MEMORY     02211000 D
                                                        % AFTER CHECKING THE PAGING BIT(BIT 22)  02212000 D
                                                        % AND THE BREAKPOINT BITS(BITS 21-20) IN 02213000 D
                                                        % THE SR#1 PORTION OF THE FSW.  BR2       02214000 D
                                                        % CONTAINS THE ADDRESS TO BE WRITTEN      02215000 D
                                                        % INTO AND MIR CONTAINS THE INFO TO BE    02216000 D
                                                        % WRITTEN.                                02217000 D
```

```
061F  4909 0000 0030 00F0          SET LC1                 % FLAG USED TO IDENTIFY THIS ROUTINE     02218000 D
0620  4809 0000 0030 00F0          BMI                     % TRANSFER MIR TO B                      02219000 D
0621  4809 0C40 2000 00F0          B = A2                  % TRANSFER B(MIR) TO A2                   02220000 D
0622  4809 A000 8B00 00F0          A1 R = B                                                         02221000 D
0623  A000 0000 0000 0020          22 = SAR                                                         02222000 D
0624  4809 0C40 0030 00F0          B                       % SEND B TO THE ADDER                     02223000 D
0625  5808 0000 0000 C0F0          IF LST THEN STEP ELSE SKIP  % CHECK IF PAGING SELECTED           02224000 D
0626  9C1C 00C0 0030 0040          PAGING - 1 = MPCR       % JUMP TO PAGING ROUTINE                 02225000 D
0627  4809 A000 8B00 C0F0   PAGEOUT: A1 R = B                                                        02226000 D
0628  9000 00C0 C030 0020          21 = SAR                                                         02227000 D
0629  4809 0C40 0030 00F0          B                       % TEST IF BREAKPOINT SELECTED            02228000 D
062A  5806 00C0 0030 C0F0          IF LST THEN STEP ELSE SKIP                                       02229000 D
062B  5FFC 0000 0030 C040          BKTCMP - 1 = MPCR       % JUMP TO BKPT ANALYSIS ROUTINE          02230000 D
062C  4809 00C0 0000 24F0   BKPTOUT: ASE                   % REFERENCE MAR2                         02231000 D
062D  4809 0F40 80LC 00F0          BMAR R = MAR2           % MAR2 CONTAINS ADDRESS                  02232000 D
062E  0000 0000 0030 0010          8 = SAR                                                          02233000 D
062F  4809 C000 0090 00F0          A2 = MIR                % RESTORE MIR                            02234000 D
0630  9809 00C0 0090 1CF0          MW2; IF SAI             % PERFORM WRITE INTO MEMORY              02235000 D
0631  9C08 00C0 C030 00F0          WHEN SAI THEN 0; STEP                                            02236000 D
0632  2809 0000 0030 00FC          IF LC1                  % CLEAR LC1                              02237000 D
0633  482D 0000 0030 00FC          JUMP                                                             02238000 D
                                                           %                                        02239000 D
                                                           %                                        02240000 D
                            IFETCH:                        % THIS ROUTINE WILL FETCH AN INSTRUCTION 02241000 D
                                                           % AND ENABLE A TRACE OF ALL REGISTERS IF 02242000 D
                                                           % PRESELECTED VIA A TRACE CONTROL CARD   02243000 D
0634  4809 0640 2030 00F0          AMPCR = A2              % STORE RETURN ADDRESS                   02244000 D
0635  4809 2000 00C0 00F0          LIT = MAR2                                                       02245000 D
0636  022C 00C0 0030 00E0          STATUS2 = LIT                                                    02246000 D
0637  2F30 00C0 0030 0060          EINPUT - 1 = CPCR       % READ IN SR #2                          02247000 D
0638  4809 0C40 0000 00FC          B                       % SR # 2 TO THE ADDER                    02248000 D
0639  4800 00C0 0000 00F0          IF MST THEN STEP ELSE SKIP  % TRACE IF TRUE                      02249000 D
063A  27E0 00C0 0030 0060          DUMPREG - 1 = CPCR      %JUMP TO REGISTER DUMPING ROUTINE        02250000 D
063B  4809 A001 0B30 00F0          A1 L = B                % ISOLATE ADDRESS IN PAR FIELD OF PSW     02251000 D
063C  0000 00C0 0030 0020          COMP 16 = SAR                                                    02252000 D
063D  4809 0C40 8B00 00F0          B R = B                                                          02253000 D
063E  4809 0C41 0010 00FC          B L = BR2               % INSERT ADDRESS INTO BR2                 02254000 D
063F  0000 00C0 0030 C030          COMP 8 = SAR            % BR2 IS FILLED FROM 2ND LSB OF SOURCE   02255000 D
0640  60E0 0000 0000 C060          EMULIN - 1 = CPCR       % RETRIEVE INSTRUCTION AT ADDRESS IN BR2 02256000 D
0641  4809 00C0 0030 C0FC          A2 = AMPCR              % RESTORE RETURN ADDRESS                 02257000 D
0642  4809 A001 2000 00F0          A1 L = A2               % ISOLATE PAR INTO A2                    02258000 D
0643  0000 00C0 0030 C020          COMP 16 = SAR                                                    02259000 D
0644  4809 C000 A030 00F0          A2 R = A2                                                        02260000 D
0645  4809 C0C0 A000 00F0          A2 + 1 R = A2           % TEST FOR UPPER MEMORY BOUNDRY          02261000 D
0646  4809 C018 0030 C0F0          A2 GTR C                                                         02262000 D
0647  7429 A0C0 4000 C0F0          IF FALSE THEN A1 + 1 = A1; JUMP  % NORMAL INCREMENT              02263000 D
0648  4809 0C40 0030 00F0          B = MIR                                                          02264000 D
0649  4809 00C1 0B00 00F0          1 L = B                 % CREATE 1024 IN B                       02265000 D
064A  A000 0000 0000 C020          COMP 10 = SAR                                                    02266000 D
064B  4809 A0C0 C000 C0F0          A1 R = A1                                                        02267000 D
064C  0000 0000 0030 0020          16 = SAR                                                         02268000 D
064D  4809 A001 4000 00F0          A1 L = A1               % CLEAR PAR                              02269000 D
064E  4809 AC0C 4030 00F0          A1 OR B = A1,BMI        % RESTORE B AND SET PAR TO 1024          02270000 D
064F  482D 00C0 C000 00F0          JUMP                                                             02271000 D
                                                           %                                        02272000 D
                                                           %                                        02273000 D
                            PAGING:                        % THIS ROUTINE WILL CALCULATE AN ACTUAL  02274000 D
                                                           % ADDRESS WHEN PASSED A PAGE NUMBER      02275000 D
                                                           % REFERENCE AND AN OFFSET IN THE BR2 REG.02276000 D
                                                           % A1,A2,A3,AMPCR MUST BE SAVED TO BE USED02277000 D
```

```
0650  4809 00C1 0000 00F0          1 L = B                  % GENERATE 512 IN B                  02278C00 D
0651  B000 0000 0000 0020          COMP 9 = SAR                                                 02279C00 D
0652  4809 2C46 0B0C 00F0          LIT + B + 1 = B          % B NOW CONTAINS 768(ADDR OF PAGEREG) 02280C00 D
0653  0FFC 0003 0C00 C0E0          255 = LIT                                                    02281C00 D
0654  4809 0CC0 0030 24F0          ASE                      % REFERENCE BR2                     02282C00 D
0655  4809 0F42 8035 00F0          NOT BMAR R = CTR         % SAVE MAR(OFFSET) IN COUNTER        02283C00 D
0656  0000 00C0 0C00 001C          B = SAR                                                      02284C00 D
0657  4809 0F40 801C 00F0          BMAR R = MAR2            % PUT BR2(PAGE ADDR REG NO.) IN MAR2 02285C00 D
0658  0000 00C0 0C00 0020          16 = SAR                                                     02286C00 D
0659  4809 C0C0 0030 00F0          A2 = MIR                 % SAVE A2 IN MIR                     02287C00 D
065A  4809 0F40 2000 00F0          BMAR  = A2               % USE A2 AS X-SELECT HOLDER FOR BMAR 02288C00 D
065B  4809 CC40 001C 00F0          A2 + B = MAR2            % MAR2 CONTAINS THE BASE ADDRESS OF THE 02289000 D
                                                            % DESIRED PAGE ADDRESS REGISTER      02290C00 D
065C  A809 C000 0000 0CF0          MR2; IF RDC              % READ CONTENTS OF PAGE ADDR. REG.   02291000 D
065D  AC08 0000 0C00 00F0          WHEN RDC THEN REX        % READ COMPLETE, STORE IN B          02292C00 D
065E  4809 0C40 2000 0CF0          B = A2                   % SAVE PAGE ADDRESS REG CONTENTS IN A2 02293C00 D
065F  4809 18C0 8B00 00F0          B100 R = B               % SET PAGE MODIFICATION BIT (BIT 15) 02294C00 D
0660  0C0C 0CC0 0000 0020          16 = SAR                                                     02295C00 D
0661  2BCB C000 C00C 00FC          IF LC1 THEN SET LC1; STEP ELSE SKIP  % STORE TO MEMORY, SET  02296000 D
                                                            % PAGE REFERENCED BIT IN PAGE ADDR REG 02297C00 D
0662  4809 CC5C 2000 00F0          A2 OR B = A2             % FORM PAGE ADDRESS REG PLUS MOD. BIT 02298C00 D
0663  4809 C000 0B30 00F0          A2 = MIR,BMI             % SAVE RETURN IN B                   02299C00 D
                                                            % MODIFIED PAGE ADDR. REG. IN MIR    02300C00 D
0664  9809 0000 0030 1CFC          MW2; IF SAI              % WRITE OUT MODIFIED PAGE REG        02301C00 D
0665  9C08 00C0 0000 00F0          WHEN SAI THEN 0                                              02302C00 D
0666  4809 C156 2C00 00F0          A2 AND LIT = A2          % ISOLATE PAGE NUMBER                02303C00 D
0667  0FFC 0003 0030 00E0          255 = LIT                % MASK OFF LOWER 8 BITS              02304C00 D
0668  4809 C0C0 0030 C0F0          A2 = MIR                 % RESTORE A2                         02305C00 D
0669  4809 0C40 2D0C 00F0          B = A2,BMI               % B HAS THE PAGE REG. CONTENTS       02306C00 D
066A  4809 6C50 8010 00F0          CTR OR B C = BR2         % BR2 CONTAINS DIRECT ADDRESS (PAGING 02307C00 D
066B  0000 0CC0 0C00 0020          16 = SAR                 % COMPLETED)                         02308C00 D
066C  2BCB C000 CC00 00FC          IF LC1 THEN SET LC1; STEP  ELSE SKIP  % ID CALLING ROUTINE   02309000 D
066D  5260 C0C0 0030 0040          PAGEOUT - 1 = MPCR       % RETURN TO EMULOUT                 02310C00 D
066E  6130 C000 0000 C040          PAGEIN - 1 = MPCR        % RETURN TO EMULIN                  02311C00 D
                                                            %                                    02312C00 D
                                                            %                                    02313C00 D
                                                            %                                    02314C00 D
                                % * * * * * * * * * * * * OPCODE IMPLEMENTATION * * * * * * * * * * * * 02315C00 D
                                                            %                                    02316C00 D
                                OPCODE:                     % THIS ROUTINE WILL FETCH AN INSTRUCTION 02317C00 D
                                                            % ANALYZE THE OPCODE(UPPER  6 BITS)   AND02318C00 D
                                                            % SELECT THE CORRESPONDING ROUTINE FOR  02319000 D
                                                            % THE SELECTED OPERATION.             02320C00 D
066F  4809 A0C0 0000 00F0          A1                       % A1 TO THE ADDER                     02321C00 D
0670  480B 0000 C000 00F0          IF MST THEN STEP ELSE SKIP  % EXAMINE REMOTE EXECUTE BIT      02322C00 D
0671  5410 0000 0030 C06C          RESTPSW - 1 = CPCR                                           02323C00 D
0672  6330 0CC0 0030 006C          IFETCH - 1 = CPCR        % GET THE NEXT INSTRUCTION           02324C00 D
0673  2809 CC00 0030 00F0          IF LC1                   % CLEAR CONDITION BITS               02325C00 D
0674  3809 0000 0C30 00F0          IF LC2                                                       02326C00 D
0675  4809 0C40 1000 00F0          B = A3                   % SAVE INSTRUCTION INTO A3           02327C00 D
0676  4809 0C40 A00C 00FC          B R = A2                 % PLACE THE 6 OPCODE BITS IN LOW ORDER 6 02328C00 D
0677  2000 0000 0030 0010          10 = SAR                 % BITS OF A2                         02329C00 D
0678  4809 C640 0040 C0FC          A2 + AMPCR = AMPCR       % CREATE ADDRESS OF OPERATION ROUTINE BY 02330000 D
0679  0C20 00C0 0C0C C0C0          OPTABLE - 1 = AMPCR      % BASE ADDRESS OF OPTABLE            02331C00 D
067A  4809 0000 0030 00F0          STEP                     % ADDING OPCODE VALUE TO OPTABLE BASE 02332C00 D
067B  4824 00C0 0000 00F0          EXEC                     % JUMP TO AMPCR + 1                  02333C00 D
                                                            %                                    02334C00 D
                                                            %                                    02335C00 D
                                OPTABLE:                    % THE FOLLOWING 64 INSTRUCTIONS CORRES- 02336C00 D
                                                            % POND TO SUBROUTINE CALLS TO THE     02337C00 D
```

```
                                                                                  % A1 CONTAINS THE FSW                                  02339C00 D
                                                                                  % A2 SERVES AS A TEMPORARY REGISTER                    02340C00 D
                                                                                  % FOR RETURNS, ETC.                                    02341C00 D
                                                                                  % P IS THE WORKING REGISTER                            02342C00 D
                                                                                  % A3 CONTAINS THE INSTRUCTION                          02343C00 D
067C  0C30 0000 0000 0040          OPCODE00 - 1 = MFCR                                                                                  02344C00 D
067D  0C40 0000 0000 0040          OPCODE01 - 1 = MFCR                                                                                  02345C00 D
067E  0C50 0000 0000 0040          OPCODE02 - 1 = MFCR                                                                                  02346C00 D
067F  0C60 0000 0000 0040          OPCODE03 - 1 = MFCR                                                                                  02347C00 D
0680  0C70 0000 0000 0040          OPCODE04 - 1 = MFCR                                                                                  02348C00 D
0681  0C80 0000 0000 0040          OPCODE05 - 1 = MFCR                                                                                  02349C00 D
0682  0C90 0000 0000 0040          OPCODE06 - 1 = MFCR                                                                                  02350C00 D
0683  0CA0 0000 0000 0040          OPCODE07 - 1 = MFCR                                                                                  02351C00 D
0684  0CB0 0000 0000 0040          OPCODE10 - 1 = MFCR                                                                                  02352C00 D
0685  0CC0 0000 0000 0040          OPCODE11 - 1 = MFCR                                                                                  02353C00 D
0686  0CD0 0000 0000 0040          OPCODE12 - 1 = MFCR                                                                                  02354C00 D
0687  0CE0 0000 0000 0040          OPCODE13 - 1 = MFCR                                                                                  02355C00 D
0688  0CF0 0000 0000 0040          OPCODE14 - 1 = MFCR                                                                                  02356C00 D
0689  0D00 0000 0000 0040          OPCODE15 - 1 = MFCR                                                                                  02357C00 D
068A  0D10 0000 0000 0040          OPCODE16 - 1 = MFCR                                                                                  02358C00 D
068B  0D20 0000 0000 0040          OPCODE17 - 1 = MFCR                                                                                  02359C00 D
068C  0D30 0000 0000 0040          OPCODE20 - 1 = MFCR                                                                                  02360C00 D
068D  0D40 0000 0000 0040          OPCODE21 - 1 = MFCR                                                                                  02361C00 D
068E  0D50 0000 0000 0040          OPCODE22 - 1 = MFCR                                                                                  02362C00 D
068F  0D60 0000 0000 0040          OPCODE23 - 1 = MFCR                                                                                  02363C00 D
0690  0D70 0000 0000 0040          OPCODE24 - 1 = MFCR                                                                                  02364C00 D
0691  0D80 0000 0000 0040          OPCODE25 - 1 = MFCR                                                                                  02365C00 D
0692  0D90 0000 0000 0040          OPCODE26 - 1 = MFCR                                                                                  02366C00 D
0693  0DA0 0000 0000 0040          OPCODE27 - 1 = MFCR                                                                                  02367C00 D
0694  0DB0 0000 0000 0040          OPCODE30 - 1 = MFCR                                                                                  02368C00 D
0695  0DC0 0000 0000 0040          OPCODE31 - 1 = MFCR                                                                                  02369C00 D
0696  0DD0 0000 0000 0040          OPCODE32 - 1 = MFCR                                                                                  02370C00 D
0697  0DE0 0000 0000 0040          OPCODE33 - 1 = MFCR                                                                                  02371C00 D
0698  0DF0 0000 0000 0040          OPCODE34 - 1 = MFCR                                                                                  02372C00 D
0699  0E00 0000 0000 0040          OPCODE35 - 1 = MFCR                                                                                  02373C00 D
069A  0E10 0000 0000 0040          OPCODE36 - 1 = MFCR                                                                                  02374C00 D
069B  0E20 0000 0000 0040          OPCODE37 - 1 = MFCR                                                                                  02375C00 D
069C  0E30 0000 0000 0040          OPCODE40 - 1 = MFCR                                                                                  02376C00 D
069D  0E40 0000 0000 0040          OPCODE41 - 1 = MFCR                                                                                  02377C00 D
069E  0E50 0000 0000 0040          OPCODE42 - 1 = MFCR                                                                                  02378C00 D
069F  0E60 0000 0000 0040          OPCODE43 - 1 = MFCR                                                                                  02379C00 D
06A0  0E70 0000 0000 0040          OPCODE44 - 1 = MFCR                                                                                  02380C00 D
06A1  0E80 0000 0000 0040          OPCODE45 - 1 = MFCR                                                                                  02381C00 D
06A2  0E90 0000 0000 0040          OPCODE46 - 1 = MFCR                                                                                  02382C00 D
06A3  0EA0 0000 0000 0040          OPCODE47 - 1 = MFCR                                                                                  02383C00 D
06A4  0EB0 0000 0000 0040          OPCODE50 - 1 = MFCR                                                                                  02384C00 D
06A5  0EC0 0000 0000 0040          OPCODE51 - 1 = MFCR                                                                                  02385C00 D
06A6  0ED0 0000 0000 0040          OPCODE52 - 1 = MFCR                                                                                  02386C00 D
06A7  0EE0 0000 0000 0040          OPCODE53 - 1 = MFCR                                                                                  02387C00 D
06A8  0EF0 0000 0000 0040          OPCODE54 - 1 = MFCR                                                                                  02388C00 D
06A9  0F00 0000 0000 0040          OPCODE55 - 1 = MFCR                                                                                  02389C00 D
06AA  0F10 0000 0000 0040          OPCODE56 - 1 = MFCR                                                                                  02390C00 D
06AB  0F20 0000 0000 0040          OPCODE57 - 1 = MFCR                                                                                  02391C00 D
06AC  0F30 0000 0000 0040          OPCODE60 - 1 = MFCR                                                                                  02392C00 D
06AD  0F40 0000 0000 0040          OPCODE61 - 1 = MFCR                                                                                  02393C00 D
06AE  0F50 0000 0000 0040          OPCODE62 - 1 = MFCR                                                                                  02394C00 D
06AF  0F60 0000 0000 0040          OPCODE63 - 1 = MFCR                                                                                  02395C00 D
06B0  0F70 0000 0000 0040          OPCODE64 - 1 = MFCR                                                                                  02396C00 D
06B1  0F80 0000 0000 0040          OPCODE65 - 1 = MFCR                                                                                  02397C00 D
```

```
06B2  0F9C 0000 0030 0040          OPCODE66 - 1 = MPCR                                  02398000 D
06B3  0FAC 0000 0030 0040          OPCODE67 - 1 = MPCR                                  02399000 D
06B4  0FBC 0000 0000 0040          OPCODE70 - 1 = MPCR                                  02400000 D
06B5  0FCC 0000 0030 0040          OPCODE71 - 1 = MPCR                                  02401000 D
06B6  0FDC 0000 0030 0040          OPCODE72 - 1 = MPCR                                  02402000 D
06B7  0FE0 0000 0000 0040          OPCODE73 - 1 = MPCR                                  02403000 D
06B8  0FFC 0000 0030 0040          OPCODE74 - 1 = MPCR                                  02404000 D
06B9  1000 0000 0000 0040          OPCODE75 - 1 = MPCR                                  02405000 D
06BA  1010 0000 0000 0040          OPCODE76 - 1 = MPCR                                  02406000 D
06BB  1020 0000 0030 0040          OPCODE77 - 1 = MPCR                                  02407000 D
                                                                  %                     02408000 D
                                                                  %                     02409000 D
                                   OPCODE00:                      % THIS ROUTINE ANALYZES 00 OPCODE   02410000 D
06BC  5F90 0000 0030 0060          XFCODE - 1 = CPCR              % "F" INTO A2                     02411000 D
06BD  4809 C640 0040 00F0          A2 + AMPCR = AMPCR             % JUMP TO DESIRED SUBFUNCTION     02412000 D
06BE  1030 0000 0000 00C0          OP00F - 1 = AMPCR              % SET UP BASE ADDR OF TYPE00 INST. 02413000 D
06BF  4809 0000 0030 00F0          STEP                                                            02414000 D
06C0  4824 0000 0030 00F0          EXEC                                                            02415000 D
                                                                  %                     02416000 D
                                                                  %                     02417000 D
06C1  4E50 0000 0030 0040          OP00F:  NOTIMP - 1 = MPCR                                  02418000 D
06C2  3000 0000 0000 0040          FAULT - 1 = MPCR                                           02419000 D
06C3  3000 0000 0000 0040          FAULT - 1 = MPCR                                           02420000 D
06C4  1040 0000 0000 0040          OP003 - 1 = MPCR                                           02421000 D
                                                                  %                     02422000 D
                                                                  %                     02423000 D
                                   OP003:                         % RX TYPE; BYTE                   02424000 D
                                                                  % THIS ROUTINE STORES THE DESIGNATED BYTE02425000 D
                                                                  % OF Y INTO BITS 0-7 OF R(A), AND CLEARS 02426000 D
                                                                  % THE REMAINING BITS OF R(A)      02427000 D
06C5  4849 0000 0000 00F0          SET LC2                        % SET FLAG FOR BYTE INST.         02428000 D
06C6  5700 0000 0030 0060          RXMFIELD - 1 = CPCR            % B = Y                           02429000 D
06C7  4809 0041 0010 00F0          B L = BR2                                                       02430000 D
06C8  0000 0000 0030 0030          COMP 8 = SAR                   % ADDR. OF Y IN BR2               02431000 D
06C9  60E0 0000 0030 0060          EMULIN - 1 = CPCR              % B = (Y)                         02432000 D
06CA  300B 0000 0000 00F0          IF NOT LC2 THEN STEP ELSE SKIP  % MS BYTE OF (Y)                 02433000 D
06CB  4809 0040 8B30 00F0          B R = B                        % OR LS BYTE OF (Y)               02434000 D
06CC  0FFC 0000 0000 0090          8 = SAR; 255 = LIT                                              02435000 D
06CD  4809 2056 0B30 00F0          LIT AND B = B;MIR              % NEW CONTENTS OR R(A)            02436000 D
06CE  2000 0000 0000 0060          SETCCA - 1= CPCR               % ARTHMETIC CC SETTING (ENTER WITH B) 02437000 D
06CF  4809 E000 8B00 00F0          A3 R = B                       % ISOLATE R(A) IN LS 4 BITS       02438000 D
06D0  80FC 0000 0030 0080          4 = SAR; 15 = LIT                                               02439000 D
06D1  4809 2056 0B30 00F0          LIT AND B = 8                                                   02440000 D
06D2  5100 0000 0000 0060          REGSTACK - 1 = CPCR                                             02441000 D
06D3  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                              02442000 D
06D4  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                               02443000 D
                                                                  %                     02444000 D
                                                                  %                     02445000 D
                                   OPCODE01:                      % THIS ROUTINE WILL ANALYZE 01 OPCODES. 02446000 D
                                                                  % THE INSTRUCTION MUST BE PASSED VIA THE 02447000 D
                                                                  % B REGISTER.                     02448000 D
06D5  5F90 0000 0030 0060          XFCODE - 1 = CPCR              % "F" INTO A2                     02449000 D
06D6  4809 C640 0040 00F0          A2 + AMPCR = AMPCR             % ADDR OF DESIRED SUBFUNCTION OF OPCODE 02450000 D
06D7  1050 0000 0000 00C0          OP01F - 1 = AMPCR              % SET UP BASE ADDR OF TYPE 01 INSTRUCTION02451000 D
06D8  4809 0000 0030 00F0          STEP                           % 01 IS DETERMINED BY ADDING THE FUNCTION02452000 D
06D9  4824 0000 0030 00F0          EXEC                           % FIELD CODE.                    02453000 D
                                                                  %                     02454000 D
                                                                  %                     02455000 D
06DA  1060 0000 0000 0040          OP01F:  OP010 - 1 = MPCR       % THIS ROUTINE JUMPS TO THE APPROPRIATE 02456000 D
06DB  1070 0000 0030 0040          OP011 - 1 = MPCR               % SUBFUNCTION FOR OPCODE 01       02457000 D
```

```
06DC  1080 0000 0000 0040          OP012 - 1 = MPCR                                    02458C00 D
06DD  1090 0000 0030 0040          OP013 - 1 = MPCR                                    02459C00 D
                                                          %                            02460C00 D
                                                          %                            02461C00 D
                              OP010:                      % RR TYPE INSTRUCTION        02462C00 D
                                                          % CONTENTS OF R(M) STORED INTO R(A)  02463C00 D
06DE  4809 2C56 0B0C 00FC          B AND LIT = B           % SELECT LS 4 BITS OF B ("P" FIELD)  02464C00 D
06DF  00F0 0000 0070 00E0          15 = LIT                                            02465C00 D
06E0  51CC 0000 0C00 0060          REGSTACK - 1 = CPCR     % SELECT REGISTER STACK TO BE USED  02466C00 D
                                                          % ADDR OF GEN REG STACK RETURNED IN MAR2  02467C00 D
06E1  2F30 0000 0000 0060          EINPUT - 1 = CPCR       % PUT CONTENTS OF R(M) IN B  02468C00 D
06E2  4809 0C40 0080 00F0          B = MIR                 % CONTENTS OF R(M) IN MIR    02469C00 D
06E3  200C 0000 0C0C C06C          SETCCA - 1= CPCR        % ARTHMETIC CC SETTING (ENTER WITH B)  02470C00 D
06E4  4809 E0C0 9000 00F0          A3 R = A3                                           02471C00 D
06E5  80FC 0000 0000 0080          4 = SAR; 15 = LIT                                   02472C00 D
06E6  4809 E155 0B00 00FC          A3 AND LIT = B          % OBTAIN "A" FIELD          02473C00 D
06E7  51CC 0000 0070 006C          REGSTACK - 1 = CPCR     % SELECT REGISTER STACK TO BE USED  02474C00 D
                                                          % ADDR OF GEN REG STACK RETURNED IN MAR2  02475C00 D
06E8  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR      % CONTENTS OF R(M) INTO R(A)  02476C00 D
06E9  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                   02477C00 D
                                                          %                            02478C00 D
                                                          %                            02479C00 D
                              OP011:                      % RI TYPE II, (Y*) INTO R(A)  02480C00 D
06EA  4809 2C56 0B00 00F0          B AND LIT = B           % OBTAIN THE "M" FIELD       02481C00 D
06EB  00F0 0000 0070 00E0          15 = LIT                                            02482C00 D
06EC  51CC 0000 0000 0060          REGSTACK - 1 = CPCR     % SELECT REGISTER STACK TO BE USED  02483C00 D
                                                          % ADDR OF GEN REG STACK RETURNED IN MAR2  02484C00 D
06ED  2F30 0000 0070 0060          EINPUT - 1 = CPCR       % (R(M)) = Y* PUT INTO B     02485C00 D
06EE  4809 0C41 0C10 00F0          B L = BR2               % SET UP ADDRESS OF Y* IN ER2  02486C00 D
06EF  000C 0000 0000 003C          COMP 8 = SAR                                        02487C00 D
06F0  60E0 0000 0070 0060          EMULIN - 1 = CPCR       % READ (Y*) INTO B           02488C00 D
06F1  4809 0C40 0080 00F0          B = MIR                 % (Y*) IN MIR                02489C00 D
06F2  200C 0000 0000 0060          SETCCA - 1= CPCR        % ARTHMETIC CC SETTING (ENTER WITH B)  02490C00 D
06F3  4809 E0C0 8B0C 00FC          A3 R = B                % EXTRACT "A" FIELD          02491C00 D
06F4  80FC 0000 000C 0080          4 = SAR; 15 = LIT                                   02492C00 D
06F5  4809 2C56 0B00 00FC          B AND LIT = B                                       02493C00 D
06F6  51CC 0000 0000 0060          REGSTACK - 1 = CPCR     % SELECT REGISTER STACK TO BE USED  02494C00 D
                                                          % ADDR OF GEN REG STACK RETURNED IN MAR2  02495C00 D
06F7  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR      % WRITE OUT CONTENTS OF Y* INTO R(A)  02496C00 D
06F8  66E0 0000 0070 0040          OPCODE - 1 = MPCR                                   02497C00 D
                                                          %                            02498C00 D
                                                          %                            02499C00 D
                              OP012:                      % RK TYPE INSTRUCTION        02500C00 D
                                                          % (Y) INTO R(A) IF M = 0 OR  02501C00 D
                                                          % (Y) + (R(M)) = R(A) IF M NE 0  02502C00 D
06F9  6330 0000 0000 0060          IFETCH - 1 = CPCR       % GET CONTENTS OF :Y: FIELD  02503C00 D
06FA  4809 E0C1 100C 00F0          A3 L = A3               % PREPARE A3 FOR STORAGE OF :Y: FIELD  02504C00 D
06FB  00F0 0000 0000 00A0          COMP 16 = SAR; 15 = LIT                             02505C00 D
06FC  4809 EC5C 100C 00F0          A3 OR B = A3                                        02506C00 D
06FD  4809 0000 0090 C0F0          C = MIR,BR2             % CLEAR MIR, BR2,            02507C00 D
06FE  4809 E0C0 A600 C0F0          A3 R = A2               % PLACE :M: FIELD IN LS 4 BITS OF A2  02508C00 D
06FF  4809 C156 001C 00FC          A2 AND LIT = MAR2       % PUT :M: FIELD INTO MAR     02509C00 D
0700  4809 0F52 000C 00F0          BMAR NEQ 0              % CHECK IF "M" FIELD 0       02510C00 D
0701  600B 0000 0000 C0F0          IF TRUE THEN STEP ELSE SKIP                         02511C00 D
0702  567C 0000 0000 006C          RM - 1 = CPCR           % ANALYZE R(M) AND RETURN VALUE IN MIR  02512C00 D
0703  4809 E001 2000 C0F0          A3 L = A2               % ISOLATE Y INTO A2          02513C00 D
0704  000C 0000 0000 C020          COMP 16 = SAR                                       02514C00 D
0705  4809 C000 A000 C0F0          A2 R = A2, BMI          % PUT Y INTO A2 AND (R(M)) INTO B  02515C00 D
0706  4809 CC40 0B30 00FC          A2 + B= B,MIR           % B,MIR CONTAINS FUTURE CONTENTS OF R(A)  02516C00 D
0707  200C 0000 0000 0060          SETCCA - 1= CPCR        % ARTHMETIC CC SETTING (ENTER WITH B)  02517C00 D
```

```
0708  4809 E000 8800 00F0          A3 R = B                    % PUT :A: FIELD IN LS 4 BITS OF B          02518000 D
0709  80F0 0000 0000 00A0          20 = SAR; 15 = LIT                                                    02519000 D
070A  4809 2C56 0800 00F0          LIT AND B = B               % PUT ADDR OF R(A) INTO B                  02520000 D
070B  51C0 0000 0000 0060          REGSTACK - 1 = CPCR         % SELECT REGISTER STACK TO BE USED         02521000 D
                                                               % ADDR OF GEN REG STACK RETURNED IN MAR2  02522000 D
070C  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR          % (R(A)) + (Y) INTO R(A)                   02523000 D
070D  66E0 0000 0000 0040          OPCODE - 1 = MPCR           % GET NEXT INSTRUCTION                     02524000 D
                                                               %                                          02525000 D
                                                               %                                          02526000 D
                                OP013:                         % RX TYPE INSTRUCTION                       02527000 D
                                                               % (Y) INTO R(A)                            02528000 D
070E  5700 0000 0000 0060          RXMFIELD - 1 = CPCR         % ANALYZE "M" FIELD CONTENTS               02529000 D
070F  4809 0041 0010 00F0          B L = BR2                   % ISOLATE Y INTO BR2                       02530000 D
0710  0000 0000 0000 0030          COMP B = SAR                                                          02531000 D
0711  60E0 0000 0000 0060          EMULIN - 1 = CPCR           % (Y) INTO B                              02532000 D
0712  4809 0040 0030 00F0          B = MIR                     % STORE (Y) INTO MIR                       02533000 D
0713  2000 0000 0000 0060          SETCCA - 1= CPCR            % ARTHMETIC CC SETTING (ENTER WITH B)      02534000 D
0714  4809 E000 8800 00F0          A3 R = B                                                              02535000 D
0715  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                                     02536000 D
0716  4809 2C56 0800 00F0          LIT AND B = B               % ISOLATE :A: FIELD                       02537000 D
0717  51C0 0000 0000 0060          REGSTACK - 1 = CPCR         % SELECT REGISTER STACK TO BE USED         02538000 D
                                                               % ADDR OF GEN REG STACK RETURNED IN MAR2  02539000 D
0718  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                                    02540000 D
0719  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                                     02541000 D
                                                               %                                          02542000 D
                                                               %                                          02543000 D
                                OPCODE02:                      % THIS ROUTINE  DECODES THE 02 OPCODE       02544000 D
071A  2809 0000 0000 00F0          IF LC1                                                                02545000 D
071B  3809 0000 0000 00F0          IF LC2                      % CLEARS LOCAL COND. BITS                  02546000 D
071C  5F90 0000 0000 0060          XFCODE - 1 = CPCR                                                     02547000 D
071D  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                                    02548000 D
071E  10A0 0000 0000 00C0          OP02F - 1 = AMPCR                                                     02549000 D
071F  4809 0000 0000 00F0          STEP                                                                  02550000 D
0720  4824 0000 0000 00F0          EXEC                                                                  02551000 D
                                                               %                                          02552000 D
0721  1080 0000 0000 0040       OP02F:  OP020 - 1 = MPCR                                                 02553000 D
0722  10C0 0000 0000 0040          OP021 - 1 = MPCR                                                      02554000 D
0723  3000 0000 0000 0040          FAULT - 1 = MPCR                                                      02555000 D
0724  10D0 0000 0000 0040          OP023 - 1 = MPCR                                                      02556000 D
                                                               %                                          02557000 D
                                OP020:                         % UNARY FUNCTION DESCRIPTOR                02558000 D
                                                               % :M: FIELD  SELECTS UNARY INST.           02559000 D
0725  4809 0040 8800 00F0          B R = B                     % ISOLATE :A: FIELD                        02560000 D
0726  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                                     02561000 D
0727  4809 2C56 0800 00F0          LIT AND B = B                                                         02562000 D
0728  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                                                   02563000 D
0729  2F30 0000 0000 0060          EINPUT - 1 = CPCR           % B = (R(A)); MAR2 = RA                   02564000 D
072A  4809 E156 2000 00F0          A3 AND LIT = A2                                                       02565000 D
072B  00F0 0000 0000 00EC          15 = LIT                    % ISOLATE :M: FIELD                       02566000 D
072C  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                                    02567000 D
072D  10E0 0000 0000 00C0          OP020M - 1 = AMPCR                                                    02568000 D
072E  4809 C15E 0000 00F0          A2 GEQ LIT                  % TEST FOR NOT IMPLEMENTED BOUNDRY        02569000 D
072F  00C0 0000 0000 00E0          12 = LIT                                                              02570000 D
0730  780B 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP                                           02571000 D
0731  3000 0000 0000 0040          FAULT - 1 = MPCR                                                      02572000 D
0732  4824 0000 0000 00F0          EXEC                                                                  02573000 D
                                                               %                                          02574000 D
0733  10F0 0000 0000 0040       OP020M: OP02000 - 1 = MPCR                                               02575000 D
0734  1100 0000 0000 0040          OP02001 - 1 = MPCR                                                    02576000 D
0735  1110 0000 0000 0040          OP02002 - 1 = MPCR                                                    02577000 D
```

```
0736  3000 0000 0000 0040          FAULT - 1 = MPCR                                  02578000 D
0737  1120 0000 0000 0040          OP02004 - 1 = MPCR                                02579000 D
0738  1130 0000 0000 0040          OP02005 - 1 = MPCR                                02580000 D
0739  1140 0000 0000 0040          OP02006 - 1 = MPCR                                02581000 D
073A  3000 0000 0000 0040          FAULT - 1 = MPCR                                  02582000 D
073B  1150 0000 0000 0040          OP02010 - 1 = MPCR                                02583000 D
073C  1160 0000 0000 0040          OP02011 - 1 = MPCR                                02584000 D
073D  1170 0000 0000 0040          OP02012 - 1 = MPCR                                02585000 D
073E  1180 0000 0000 0040          OP02013 - 1 = MPCR                                02586000 D
                                                   %                                 02587000 D
                                OP02000:           % TYPE RR; MAKE POSITIVE           02588000 D
                                                   % IF (R(A)) < 0, (R(A)): -> RA     02589000 D
                                                   % IF (R(A)) => 0, (R(A)) UNCHANGED 02590000 D
073F  4809 0041 2830 00F0          B L = A2,B                                        02591000 D
0740  000C 0003 0000 0020          COMP 16 = SAR      % SHIFT (R(A)) TO UHW OF A2,B   02592000 D
0741  4809 0812 0000 00F0          A2 EQL B100        % TEST FOR MAX NEG. NUMBER      02593000 D
0742  6BC9 0003 0000 00F0          IF TRUE THEN SET LC1                              02594000 D
0743  5C20 0000 0000 0060          OVBIT - 1 = CPCR   % SET OR CLEAR OV BIT          02595000 D
0744  4809 0003 0000 00F0          STEP                                              02596000 D
0745  0000 0003 0000 0020          16 = SAR                                          02597000 D
0746  4809 C003 0000 00F0          A2                 % A2 TO THE ADDER              02598000 D
0747  400B 0003 0000 00F0          IF NOT MST THEN STEP ELSE SKIP                    02599000 D
0748  66E0 0003 0000 0040          OPCODE - 1 = MPCR                                 02600000 D
0749  4849 0C5E 8030 00F0          -B R = B, MIR; SET LC2                            02601000 D
074A  7BC9 0003 0030 00F0          IF AOV THEN SET LC1    % CHECK FOR CARRY          02602000 D
074B  1E70 0003 0030 0060          CARRY - 1 = CPCR                                  02603000 D
074C  380B 0003 0030 00F0          IF LC2 THEN STEP ELSE SKIP  % LC2 INDICATES WHETHER RA 02604000 D
074D  2F50 0003 0000 0060          EOUTPUT - 1 = CPCR     % MUST BE UPDATED          02605000 D
074E  2000 0003 0000 0060          SETCCA - 1 = CPCR      % B CONTAINS (RA)          02606000 D
074F  66E0 0003 0000 0040          OPCODE - 1 = MPCR                                 02607000 D
                                                   %                                 02608000 D
                                OP02001:           % TYPE RR; MAKE NEGATIVE           02609000 D
                                                   % IF (R(A)) > 0, (R(A)): -> RA     02610000 D
                                                   % IF (R(A)) <= 0, (R(A)) UNCHANGE  02611000 D
0750  4809 0041 2830 00F0          B L = A2,B         % (R(A)) INTO UHW OF A2,B       02612000 D
0751  0000 0003 0030 0020          COMP 16 = SAR                                     02613000 D
0752  4809 C000 0030 00F0          A2                 % A2 TO THE ADDER               02614000 D
0753  4419 C012 0000 00F0          IF NOT MST THEN A2 EQL 0; SKIP                    02615000 D
0754  56E0 0003 0000 0040          OPCODE - 1 = MPCR                                 02616000 D
0755  580B 0000 0030 00F0          IF TRUE THEN STEP ELSE SKIP                       02617000 D
0756  56E0 0000 0030 0040          OPCODE - 1 = MPCR                                 02618000 D
0757  4849 0C5E 8030 00FC          -B R = B, MIR; SET LC2                            02619000 D
0758  7BC9 0000 0000 00F0          IF AOV THEN SET LC1    % CHECK FOR CARRY          02620000 D
0759  1E70 0000 0000 0060          CARRY - 1 = CPCR                                  02621000 D
075A  380B 0000 0000 00FC          IF LC2 THEN STEP ELSE SKIP                        02622000 D
075B  2F50 0000 0000 006C          EOUTPUT - 1 = CPCR     %(R(A)): -> RA ONLY IF LC2 IS SET 02623000 D
075C  2000 0000 0000 0060          SETCCA - 1 = CPCR      % B = (R(A)) OR (R(A)):    02624000 D
075D  56E0 0000 0000 0040          OPCODE - 1 = MPCR                                 02625000 D
                                                   %                                 02626000 D
                                OP02002:           % ROUND TYPE RR                    02627000 D
                                                   % IF (R(A)) >= 0 (R(A))+(R(A+1))15 -> RA 02628000 D
                                                   % IF (R(A)) < 0 (R(A))-(R(A+1))15 -> RA 02629000 D
                                                   % RA MUST BE EVEN                  02630000 D
075E  4809 0041 2000 00F0          B L = A2           % B = (R(A)); MAR2 = RA         02631000 D
075F  0010 0003 0000 00AC          COMP 16 = SAR; 1 = LIT                            02632000 D
0760  4809 2F5C 0B00 00F0          LIT OR BMAR = B        % R(A+1)                    02633000 D
0761  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2          02634000 D
0762  2F30 0000 0000 006C          EINPUT - 1 = CPCR      % B = (R(A+1))              02635000 D
0763  4809 C000 0030 00F0          A2                                                02636000 D
0764  4819 0003 0000 00FC          IF MST THEN SKIP                                  02637000 D
```

```
0765  119C 0000 0030 0040          RPOS - 1 = MPCR                                  02638000 D
                                                   %                                02639000 D
0766  4809 0C42 0B00 00F0   RNEG:   NOT B = B      % COMPLEMENT B (1:S)             02640000 D
0767  4809 0C40 8B00 00F0           B R = B        % ISOLATE BIT 15 IN B             02641000 D
0768  B00C 0000 0030 0010           15 = SAR                                        02642000 D
0769  4809 0C41 0B00 00F0           B L = B        % CLEAN OUT B REGISTER            02643000 D
076A  1000 0000 0030 0000           COMP 31 = SAR                                   02644000 D
076B  4809 0C40 8B00 00F0           B R = B        % PLACE BIT 15 IN BIT 16          02645000 D
076C  B000 0000 0030 0010           15 = SAR                                        02646000 D
076D  4849 0C5E 0030 00F0           A2 - B = MIR; SET LC2                           02647000 D
076E  7BC9 0000 0000 00F0           IF AOV THEN SET LC1  % CHECK FOR CARRY          02648000 D
076F  1E70 0000 0030 0060           CARRY - 1 = CPCR                                02649000 D
0770  9F10 0000 0030 C060           CHECKOV - 1 = CPCR   % OVERFLOW CHECK           02650000 D
0771  11A0 0000 0030 0040           END02002 - 1 = MPCR                             02651000 D
                                                   %                                02652000 D
0772  4BC9 0C41 0B00 00F0   RPOS:   B L = B        % SHIFT BIT POS 15 OF RA+1 INTO BIT 16  02653000 D
0773  8000 0000 0030 0030           COMP 1 = SAR                                    02654000 D
0774  4809 0C40 0C90 00F0           A2 + B = MIR                                    02655000 D
0775  7BC9 0000 0000 00F0           IF AOV THEN SET LC1  % CHECK FOR CARRY          02656000 D
0776  1E70 0000 0030 C060           CARRY - 1 = CPCR                                02657000 D
0777  9F10 0000 0000 006C           CHECKOV - 1 = CPCR   % CHECK FOR OVERFLOW       02658000 D
0778  4809 0000 0000 24F0   END02002:ASE                                            02659000 D
0779  4809 0F40 2090 00F0           BMAR = A2          % SET UP R(A) INTO MAR2       02660000 D
077A  4809 C0CE 001C 00F0           A2 AND B110 = MAR2, BMI % FORM RA               02661000 D
077B  4809 0C40 8B30 00F0           B R = B,MIR        % RESULT  (R(A)) STORE INTO LHW OF B,MIR  02662000 D
077C  0000 0000 0030 0020           16 = SAR                                        02663000 D
077D  2C00 0000 0000 C060           SETCCA - 1 = CPCR                               02664000 D
077E  2F50 0000 0030 0060           EOUTPUT - 1 = CPCR                              02665000 D
077F  66E0 0000 0000 0040           OPCODE - 1 = MPCR                               02666000 D
                                                   %                                02667000 D
                            OP02004:               % 2:S COMPLEMENT; TYPE RR        02668000 D
0780  4809 0C41 2B30 00F0           B L = A2,B     % A2,B HAVE (R(A)) IN UHW; MAR2 = RA  02669000 D
0781  0000 0000 0030 0020           COMP 16 = SAR                                   02670000 D
0782  4809 0B12 0000 C0F0           A2 EOL B100                                     02671000 D
0783  6BC9 0000 0000 00F0           IF TRUE THEN SET LC1 % CHECK FOR MAX NEG. NUM.  02672000 D
0784  5020 0000 0000 C06C           OVBIT - 1 = CPCR   % SET/CLEAR OVBIT            02673000 D
0785  0000 0000 0030 002C           16 = SAR                                        02674000 D
0786  4809 0C5E 8B30 00F0           - B R = B,MIR      % TWO:S COMPLEMENT (R(A))     02675000 D
0787  7BC9 0000 0000 00F0           IF AOV THEN SET LC1  % CHECK FOR CARRY          02676000 D
0788  1E70 0000 0000 0060           CARRY - 1 = CPCR                                02677000 D
0789  2000 0000 0000 0060           SETCCA - 1 = CPCR  % B = (R(A))                 02678000 D
078A  2F50 0000 0000 C06C           EOUTPUT - 1 = CPCR                              02679000 D
078B  66E0 0000 0000 004C           OPCODE - 1 = MPCR                               02680000 D
                                                   %                                02681000 D
                            OP02005:               % TWO:S COMPLEMENT DOUBLE; TYPE RR  02682000 D
                                                   % (RA,RA+1): -> RA,RA+1          02683000 D
078C  4809 0000 0000 24F0           ASE            % REF BR2                        02684000 D
078D  4809 0F41 0C20 00F0           BMAR L = BR1   % TEMP STORAGE OF R(A)           02685000 D
078E  0000 0000 0030 003C           COMP 8 = SAR                                    02686000 D
078F  4809 0C41 2C30 00F0           B L = A2       % B = (R(A)); MAR2 = RA           02687000 D
0790  001C 0000 0030 00A0           COMP 16 = SAR ; 1 = LIT                         02688000 D
0791  4809 2F5C 001C 00F0           LIT OR BMAR = MAP2   % R(A+1)                   02689000 D
0792  2F30 0000 0000 C060           EINPUT - 1 = CPCR  % B = (R(A+1)) MAR2 = RA + 1 02690000 D
0793  4809 CC5C 2B00 00F0           A2 OR 6 = A2, B    % A2 = (RA,RA+1)              02691000 D
0794  4809 0B12 0030 C0F0           A2 EOL B100                                     02692000 D
0795  6BC9 0000 0000 00FC           IF TRUE THEN SET LC1 % TEST FOR MAX NEG NUM     02693000 D
0796  5020 0000 0000 C06C           OVBIT - 1 = CPCR                                02694000 D
0797  4809 0C5E 0B00 00FC           - B = B            % TWO:S COMPLEMENT OF (R(A),R(A+1))  02695000 D
0798  7BC9 0000 0000 00F0           IF AOV THEN SET LC1  % CHECK FOR CARRY          02696000 D
0799  1E70 0000 0030 C060           CARRY - 1 = CPCR                                02697000 D
```

```
079A  4809 0C41 1000 00F0          B L = A3                 % ISOLATE (RCA)) INTO MIR                    02699000 D
079D  0000 0000 0030 C020          COMP 16 = SAR                                                        02690000 D
079C  4809 E000 8030 C0FC          A3 R = MIR                                                           02700000 D
079D  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR                                                   02701000 D
079E  4809 0C40 8030 20F0          B R = MIR; ASR           % (R(A)) INTO MIR                           02702000 D
079F  49C9 0F43 801C 00F0          BMAR R = MAR2; SET LC1   % R(A); SET DOUBLE WORD ARITH COMPARE        02703000 D
07A0  000C 0000 0030 0010          8 = SAR                                                              02704000 D
07A1  2F50 0000 000C C060          EOUTPUT - 1 = CPCR                                                   02705000 D
07A2  2000 0000 0030 0060          SETCCA - 1 = CPCR        % B CONTAINS (R(A),R(A+1))                  02706000 D
07A3  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                                    02707000 D
                                                            %                                            02708000 D
                                OP02006:                    % ONE'S COMPLEMENT; TYPE RR                   02709000 D
                                                            % (R(A)) BIT BY BIT -> RA                    02710000 D
                                                            % B = (R(A)); MAR2 = RA                      02711000 D
07A4  4809 0C43 0830 00F0          NOT B L = B              % B = (R(A))                                 02712000 D
07A5  0000 0000 0030 0020          COMP 16 = SAR                                                        02713000 D
07A6  4809 0C40 8830 C0FC          B R = B, MIR                                                         02714000 D
07A7  2F50 0000 0030 006C          EOUTPUT - 1 = CPCR                                                   02715000 D
07A8  2000 0000 0000 0060          SETCCA - 1 = CPCR                                                    02716000 D
07A9  66E0 0000 0030 C040          OPCODE - 1 = MPCR                                                    02717000 D
                                                            %                                            02718000 D
                                OP02010:                    % INCREMENT RA BY 1; TYPE RR                  02719000 D
                                                            % (R(A)) + 1 -> RA; SET CC                   02720000 D
                                                            % B = (R(A)) ; MAR2 = RA                     02721000 D
07AA  4809 0C41 2030 C0F0          B L = A2                                                             02722000 D
07AB  0000 0000 0000 0020          COMP 16 = SAR            % (R(A)) IN UHW OF A2                        02723000 D
07AC  48C9 00C1 0830 C0F0          1 L = B                  % INCREMENT  IN BIT 16                       02724000 D
07AD  4809 CC40 0030 00F0          A2 + B = MIR                                                         02725000 D
07AE  78C9 000D 0030 00F0          IF AOV THEN SET LC1      % CHECK FOR CARRY                           02726000 D
07AF  1E70 000D 0030 C060          CARRY - 1 = CPCR                                                     02727000 D
07B0  4F10 0000 0000 0060          CHECKOV - 1 = CPCR       % CHECK FOR OVERFLOW                        02728000 D
07B1  4809 0000 0000 00FC          BNI                                                                  02729000 D
07B2  4809 0C40 8830 00F0          B R = B,MIR              % RESULT IN LHW OF B,MIR                    02730000 D
07B3  0000 0000 0030 C020          16 = SAR                                                             02731000 D
07B4  2000 00C0 0000 006C          SETCCA - 1 = CPCR                                                    02732000 D
07B5  2F50 00C0 0030 0060          EOUTPUT - 1 = CPCR                                                   02733000 D
07B6  66E0 0000 0030 C040          OPCODE - 1 = MPCR                                                    02734000 D
                                                            %                                            02735000 D
                                OP02011:                    % DECREMENT RA BY 1; TYPE RR                  02736000 D
                                                            % (R(A)) - 1 -> RA                           02737000 D
                                                            % B = (R(A)); MAR2 = RA                      02738000 D
07B7  4809 0C41 2000 00F0          B L = A2                                                             02739000 D
07B8  0000 0000 0000 0020          COMP 16 = SAR            % (R(A)) IN UHW OF A2                        02740000 D
07B9  4849 00C1 0830 00F0          1 L = B; SET LC2         % DECREMENT VALUE IN BIT 16                  02741000 D
07BA  4809 CC5E 0030 00F0          A2 - B = MIR                                                         02742000 D
07BB  78C9 00C0 0030 0CFC          IF AOV THEN SET LC1      % CHECK IF CARRY GENERATED                  02743000 D
07BC  1E70 00C0 0030 0060          CARRY - 1 = CPCR                                                     02744000 D
07BD  4F10 0000 0030 006C          CHECKOV - 1 = CPCR       % CHECK IF OV GENERATED                     02745000 D
07BE  4809 00C0 0000 C0FC          BNI                                                                  02746000 D
07BF  4809 0C40 8830 00F0          B R = B,MIR              % RESULT INTO LHW OF B,MIR                  02747000 D
07C0  0000 0000 0030 0020          16 = SAR                                                             02748000 D
07C1  2000 C000 0030 0060          SETCCA - 1 = CPCR                                                    02749000 D
07C2  2F50 00C0 0000 0060          EOUTPUT - 1 = CPCR                                                   02750000 D
07C3  66E0 0000 0030 0040          OPCODE - 1 = MPCR                                                    02751000 D
                                                            %                                            02752000 D
                                OP02012:                    % INCREASE RA BY TWO; TYPE RR                 02753000 D
                                                            % (R(A)) + 2 -> RA                           02754000 D
                                                            % B = (R(A)); MAR2 = RA                      02755000 D
07C4  4809 0C41 2030 00F0          B L = A2                                                             02756000 D
07C5  0020 0000 0000 00A0          COMP 16 = SAR; 2 = LIT% (R(A)) IN UHW OF A2                          02757000 D
```

```
07C6  4809 2001 0B30 00F0          LIT L = B                                                  02758000 D
07C7  4809 CC40 0C30 00F0          A2 + B = MIR                                               02759000 D
07C8  7BC9 0000 0000 00F0          IF AOV THEN SET LC1     % CHECK FOR CARRY                  02760000 D
07C9  1E70 0000 0000 0060          CARRY - 1 = CPCR                                           02761000 D
07CA  4F10 0000 0000 0060          CHECKOV - 1 = CPCR      % CHECK FOR OVERFLOW               02762000 D
07CB  4809 0000 0000 00FC          BMI                                                        02763000 D
07CC  4809 0C40 8B30 00F0          B R = B,MIR             % RESULT IN LHW OF B,MIR           02764000 D
07CD  0000 0000 0000 0020          16 = SAR                                                   02765000 D
07CE  2000 0000 0000 0060          SETCCA - 1 = CPCR                                          02766000 D
07CF  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                         02767000 D
07D0  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                          02768000 D
                                                           %                                  02769000 D
                                OP02013:                   % DECREASE RA BY TWO; TYPE RR       02770000 D
                                                           % (RA) - 2 -> RA                   02771000 D
                                                           % B = (RA); MAR2 = RA              02772000 D
07D1  4809 0C41 2030 00F0          B L = A2                                                   02773000 D
07D2  0020 0000 0000 C0A0          COMP 16 = SAR; 2 = LIT% (R(A)) IN UHW OF A2                02774000 D
07D3  4849 2001 0B30 00F0          LIT L = B; SET LC2      % LC2 IS SUBTRACTION FLAG          02775000 D
07D4  4809 CC5E 0030 00F0          A2 - B = MIR                                               02776000 D
07D5  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                                        02777000 D
07D6  1E70 0000 0000 0060          CARRY - 1 = CPCR        % SET/CLEAR CARRY BIT              02778000 D
07D7  4F10 0000 0000 0060          CHECKOV - 1 = CPCR      % CHECK FOR OVERFLOW               02779000 D
07D8  4809 0000 0000 00F0          BMI                                                        02780000 D
07D9  4809 0C40 8B30 00F0          B R = B,MIR             % RESULT IN LHW OF B AND MIR       02781000 D
07DA  0000 0000 0000 0020          16 = SAR                                                   02782000 D
07DB  2000 0000 0000 0060          SETCCA - 1 = CPCR                                          02783000 D
07DC  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                         02784000 D
07DD  66E0 0000 0000 004C          OPCODE - 1 = MPCR                                          02785000 D
                                                           %                                  02786000 D
                                OP021:                     % LOAD DOUBLE; TYPE RI(II)         02787000 D
                                                           %(Y*,Y* + 1) -> RA,RA+1; SET CC    02788000 D
07DE  4809 2C56 0B30 00F0          B AND LIT = B           % ISOLATE :M: FIELD                02789000 D
07DF  00F0 0000 0000 00E0          15 = LIT                                                   02790000 D
07E0  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                                        02791000 D
07E1  2F30 0000 0000 0060          EINPUT - 1 = CPCR       % B = (R(M)) = Y*                  02792000 D
07E2  482C 0000 0000 0060          LDBLE - 1 = CPCR        % CALL LOAD DOUBLE ROUTINE         02793000 D
07E3  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                          02794000 D
                                                           %                                  02795000 D
                                OP023:                     % LOAD DOUBLE; TYPE RX             02796000 D
                                                           % (Y,Y+1) -> RA,RA+1               02797000 D
07E4  5700 0000 0000 0060          RXMFIELD - 1 = CPCR     % B = Y                            02798000 D
07E5  4820 0000 0000 006C          LDBLE - 1 = CPCR        % CALL LOAD DOUBLE ROUTINE         02799000 D
07E6  66E0 0000 0000 004C          OPCODE - 1 = MPCR                                          02800000 D
                                                           %                                  02801000 D
                                OPCODE03:                  % THIS ROUTINE ANALYZES THE 03 OPCODE 02802000 D
07E7  5F90 0000 0000 C060          XFCODE - 1 = CPCR                                          02803000 D
07E8  4809 C640 0040 00FC          A2 + AMPCR = AMPCR                                         02804000 D
07E9  11B0 0000 0000 00CC          OP03F - 1 = AMPCR                                          02805000 D
07EA  4809 00C0 0000 00F0          STEP                                                       02806000 D
07EB  4824 0000 0000 C0F0          EXEC                                                       02807000 D
                                                           %                                  02808000 D
07EC  11C0 0000 0000 0040       OP03F:  OP030 - 1 = MPCR                                      02809000 D
07ED  3000 0000 0000 0040          FAULT - 1 = MPCR                                           02810000 D
07EE  3000 0000 0000 0040          FAULT - 1 = MPCR                                           02811000 D
07EF  1190 0000 0000 C040          OP033 - 1 = MPCR                                           02812000 D
                                OP030:                                                        02813000 D
07F0  4809 2C56 2030 00F0          B AND LIT = A2                                             02814000 D
07F1  00FC 0000 0000 00E0          15 = LIT                                                   02815000 D
07F2  4809 C640 0040 00FC          A2 + AMPCR = AMPCR                                         02816000 D
07F3  11EC 0000 0000 00C0          OP030M - 1 = AMPCR                                         02817000 D
```

```
07F4  4809 C15E 0000 00F0          A2 GEQ LIT                                          02818000 D
07F5  0070 0000 0000 00E0          7 = LIT                                             02819000 D
07F6  780B 0000 0000 0000          IF TRUE THEN STEP ELSE SKIP                         02820000 D
07F7  4E50 0000 0030 0040          NOTIMP - 1 = MPCR      % INSTRUCTION NOT IMPLEMENTED 02821000 D
07F8  4824 0003 0000 00F0          EXEC                                                02822000 D
                                                          %                            02823000 D
                                                          %                            02824000 D
                                   OP03011:                                            02825000 D
07F9  11F0 0000 0000 0040          OP03000 - 1 = MPCR                                  02826000 D
07FA  1200 0000 0030 0040          OP03001 - 1 = MPCR                                  02827000 D
07FB  1210 0003 0000 0040          OP03002 - 1 = MPCR                                  02828000 D
07FC  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                                   02829000 D
07FD  1220 0000 0000 0040          OP03004 - 1 = MPCR                                  02830000 D
07FE  1230 0000 0000 0040          OP03005 - 1 = MPCR                                  02831000 D
07FF  1240 0000 0000 0040          OP03006 - 1 = MPCR                                  02832000 D
                                                          %                            02833000 D
                                   OP03000:               % EXECUTIVE RETURN (P) + 1 -> RA; HALT 02834000 D
0800  4809 E000 8B30 00F0          A3 R = B                                            02835000 D
0801  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                   02836000 D
0802  4809 2056 0000 00F0          LIT AND B = B          % B = 'A' FIELD              02837000 D
0803  51C0 0000 0030 0060          REGSTACK - 1 = CPCR    % RA = MAR2                  02838000 D
0804  4809 A0C1 0B30 00F0          A1 + 1 L = B           % ISOLATE PAR INTO B INCREMENTED BY 1 02839000 D
0805  0000 0000 0000 0020          COMP 16 = SAR                                       02840000 D
0806  4809 0C40 8B30 00F0          B R = B,MIR            % (P) + 1 INTO B AND MIR     02841000 D
0807  200C 0000 0030 0060          SETCCA - 1 = CPCR      % SET CONDITION CODE         02842000 D
0808  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR     % (P) + 1 -> RA              02843000 D
0809  5060 0000 0000 0040          STOPTIME - 1 = MPCR    % STOP MACHINE EXECUTION     02844000 D
                                                          %                            02845000 D
                                                          %                            02846000 D
                                   OP03001:               % STORE (SR1) INTO RA         02847000 D
080A  4809 E000 9000 00F0          A3 R = A3                                           02848000 D
080B  80F0 0000 0000 008C          4 = SAR; 15 = LIT                                   02849000 D
080C  4809 E156 0B30 00F0          A3 AND LIT = B                                      02850000 D
080D  51C0 0000 0030 0060          REGSTACK - 1 = CPCR    % RA = MAR2                  02851000 D
080E  4809 A0C0 8B30 00F0          A1 R = B,MIR           % (SR1) INTO MIR              02852000 D
080F  0000 0000 0000 0020          16 = SAR                                            02853000 D
0810  200C 0000 0000 0060          SETCCA - 1 = CPCR                                   02854000 D
0811  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR     % STORE (SR1) INTO RA        02855000 D
0812  66E0 0000 0030 0040          OPCODE - 1 = MPCR                                   02856000 D
                                                          %                            02857000 D
                                   OP03002:               % STORE (SR2) INTO RA         02858000 D
0813  4809 E000 9030 00F0          A3 R = A3                                           02859000 D
0814  80F0 0000 0000 008C          4 = SAR; 15 = LIT                                   02860000 D
0815  4809 E155 0B30 00F0          A3 AND LIT = B                                      02861000 D
0816  51C0 0000 0030 0060          REGSTACK - 1 = CPCR    % RA = MAR2                  02862000 D
0817  4809 0000 0030 24F0          ASE                                                 02863000 D
0818  4809 0F40 2030 00F0          BMAR = A2              % RA INTO A2                 02864000 D
0819  4809 20C3 001C 00FC          LIT = MAR2                                          02865000 D
081A  0220 0000 0000 00E0          STATUS2 = LIT                                       02866000 D
081B  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % B = SR2                    02867000 D
081C  4809 0C41 0B30 00F0          B L = B                % ISOLATE UYK20 SR2           02868000 D
081D  3000 0000 0030 0020          COMP 16 = SAR                                       02869000 D
081E  4809 0C40 8030 00F0          B R = MIR                                           02870000 D
081F  4809 C003 001C 00F0          A2 = MAR2              % TRANSFER RA INTO MAR2      02871000 D
0820  200C 0000 0030 0060          SETCCA - 1 = CPCR                                   02872000 D
0821  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR     % (SR2) INTO RA              02873000 D
0822  66E0 0000 0030 0040          OPCODE - 1 = MPCR                                   02874000 D
                                                          %                            02875000 D
                                                          %                            02876000 D
                                   OP03004:               % LOAD P; (R(A)) INTO P       02877000 D
```

```
0823  4809 E000 9000 00F0        A3 R = A3                                          02878000 D
0824  80F0 0000 0000 0080        4 = SAR; 15 = LIT                                  02879000 D
0825  4809 E156 0B00 00F0        A3 AND LIT = B                                     02880000 D
0826  51C0 0000 0C00 0060        REGSTACK - 1 = CPCR     % RA IN MAR2               02881000 D
0827  2F30 0000 0000 0060        EINPUT - 1 = CPCR       % B = (R(A))               02882000 D
0828  4809 A000 C000 00FC        A1 R = A1                                          02883000 D
0829  0000 0000 0000 0020        16 = SAR                                           02884000 D
082A  4809 A001 4000 00F0        A1 L = A1               % ZERO PAR                 02885000 D
082B  4809 AC5C 4000 00F0        A1 OR B = A1            % (R(A)) INTO P UPPER 16 BITS UNTOUCHED 02886000 D
082C  66E0 0000 0000 C040        OPCODE - 1 = MPCR                                  02887000 D
                                                         %                          02888000 D
                                                         %                          02889000 D
                                 OP03005:                % LOAD SR1 ; (R(A)) INTO SR1 02890000 D
082D  4809 E000 9000 00F0        A3 R = A3                                          02891000 D
082E  80F0 0000 0000 0080        4 = SAR; 15 = LIT                                  02892000 D
082F  4809 E156 0B00 00F0        A3 AND LIT = B                                     02893000 D
0830  51C0 0000 0000 C060        REGSTACK - 1 = CPCR     % RA IN MAR2               02894000 D
0831  2F30 0000 0000 0060        EINPUT - 1 = CPCR       % B = (R(A))               02895000 D
0832  4809 0C41 0090 00FC        B L = MIR               % STORE (R(A)) IN UHW OF MIR 02896000 D
0833  0000 0000 0000 0020        COMP 16 = SAR                                      02897000 D
0834  4809 2001 2000 00F0        LIT L = A2              % CREATE BIT MASK FOR LOADER TOGGLES 02898000 D
0835  3070 0000 0000 0080        7 = LIT; COMP 29 = SAR                             02899000 D
0836  4809 2001 C000 C0F0        LIT L = B               % LOWER MASK OF SR1         02900000 D
0837  0700 0000 0000 00A0        112 = LIT; COMP 16 = SAR                           02901000 D
0838  4809 CC5C 0800 C0FC        A2 OR B = B             % CREATE MASK ONES IN UHW OF B 02902000 D
0839  4809 0C42 2000 00F0        NOT B = A2, BMI         % MASK FOR (RA)            02903000 D
083A  4809 CC56 0090 00FC        A2 AND B = MIR          % MASK OFF (RA), STORE IN MIR 02904000 D
083B  4809 C002 0B00 00F0        NOT A2 = B              % MASK FOR A1 (PSW)        02905000 D
083C  4809 AC56 2000 00FC        A1 AND B = A2, BMI      % MASK OFF SAVED PART OF A1 INTO A2 02906000 D
083D  4809 CC5C C800 00F0        A2 OR B = B             % CREATE NEW A1 VALUE FROM (R(A)) 02907000 D
083E  4809 AC01 4000 C0F0        A1 L = A1               % CLEAR UHW OF A1           02908000 D
083F  0000 0000 0000 0020        COMP 16 = SAR                                      02909000 D
0840  4809 A000 C000 00FC        A1 R = A1               % RESTORE A1                02910000 D
0841  4809 AC5C 4000 00F0        A1 OR B = A1            % RESTORE A1 (ACTIVE PSW)   02911000 D
0842  66E0 0000 0000 C040        OPCODE - 1 = MPCR       % UNTOUCHED                 02912000 D
                                                         %                          02913000 D
                                 OP03006:                % LOAD SR2; (R(A)) INTO SR2 02914000 D
0843  4809 E000 9000 00F0        A3 R = A3                                          02915000 D
0844  80F0 0000 0000 0080        4 = SAR; 15 = LIT                                  02916000 D
0845  4809 E156 0B00 00F0        A3 AND LIT = B                                     02917000 D
0846  51C0 0000 0000 0060        REGSTACK - 1 = CPCR     % RA = MAR2                02918000 D
0847  2F30 0000 0000 0060        EINPUT - 1 = CPCR       % B = (R(A))               02919000 D
0848  4809 0C40 2000 00F0        B = A2                  % (R(A)) = A2               02920000 D
0849  4809 2000 0010 00F0        LIT = MAR2                                         02921000 D
084A  0220 0000 0000 00E0        STATUS2 = LIT                                      02922000 D
084B  2F30 0000 0000 0060        EINPUT - 1 = CPCR       % B = (STATUS2)            02923000 D
084C  4809 0C40 8B00 00F0        B R = B                 % ISOLATE UPPER 16 BITS OF STATUS2 02924000 D
084D  0000 0000 0000 0020        16 = SAR                                           02925000 D
084E  4809 0C41 0B00 00F0        B L = B                                            02926000 D
084F  4809 CC5C 0080 C0F0        A2 OR B = MIR           %  NEW STATUS2 CREATED     02927000 D
0850  2F50 0000 0000 C060        EOUTPUT - 1 = CPCR      % (R(A)) INTO STATUS2      02928000 D
0851  56E0 0000 0000 0040        OPCODE - 1 = MPCR                                  02929000 D
                                                         %                          02930000 D
                                                         %                          02931000 D
                                 OP033:                  % LOAD MULTIPLE, TYPE RX    02932000 D
                                                         % (Y ... Y + M - A + 1) INTO RA ... RM 02933000 D
0852  5330 0000 0000 0060        IFETCH - 1 = CPCR       % B = :Y: = Y               02934000 D
0853  4809 0C41 0B00 C0F0        B L = B                                            02935000 D
0854  00F0 0000 0000 00A0        COMP 16 = SAR; 15 = LIT                            02936000 D
0855  4809 EC5C 1000 00F0        A3 OR B = A3            % STORE Y IN UHW OF A3      02937000 D
```

```
0856  4809 E156 2000 00F0          A3 AND LIT = A2        % A2 = :M: FIELD                  02938000 D
0857  4809 E000 8000 00F0          A3 R = B                                                 02939000 D
0858  5000 00C0 0030 0000          Q = SAR                                                  02940000 D
0859  4809 2C56 0600 C0F0          LIT AND B = B          % B = :A: FIELD                   02941000 D
085A  4809 CC5E 0000 CFFC          A2 LSS B               % IF M < A THEN ADD 16            02942000 D
085B  0100 00C0 0C00 00E0          16 = LIT                                                 02943000 D
085C  7000 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP                              02944000 D
085D  4809 C140 2030 00FC          A2 + LIT = A2                                            02945000 D
085E  4809 CC5E 2000 00F0          A2 - B = A2            % SUBTRACT M FROM A               02946000 D
085F  4809 C012 0030 C0FC          A2 EQL 0               % IF M = A THEN COUNTER = 0       02947000 D
0860  6819 0000 0000 00F0          IF TRUE THEN SKIP                                        02948000 D
0861  4809 C0C0 2000 24FC          A2 + 1 = A2; ASE       % COMPUTE M - A + 1               02949000 D
0862  51C0 0000 0030 0060          REGSTACK - 1 = CPCR    % B HAS RA; MAR2 = RA             02950000 D
0863  4809 0F41 0020 40F0          BMAR L = BR1,CSAR      % SAVE RA IN BR1                  02951000 D
0864  0000 0000 0030 0030          COMP B = SAR                                             02952000 D
0865  4809 E000 9010 C0FC          A3 R = A3,BR2          % STORE Y IN BR2                  02953000 D
0866  4809 E000 9000 00F0          A3 R = A3              % STORE Y IN A3; RIGHT JUSTIFIED  02954000 D
0867  60E0 00F0 0000 0060  LDMUL:  EMULIN - 1 = CPCR      % LOOP RETURN; B = (Y)            02955000 D
0868  4809 0C40 0090 C0F0          B = MIR                % MIR = (Y)                       02956000 D
0869  4809 0000 0000 20F0          ASR                                                      02957000 D
086A  4809 0F40 8B1C 00F0          BMAR R = B,MAR2        % RA INTO B FROM BR1              02958000 D
086B  0000 0000 0000 0010          B = SAR                                                  02959000 D
086C  2F50 00C0 0000 0060          EOUTPUT - 1 = CPCR     % (Y) INTO RA                     02960000 D
086D  4809 0C46 0B00 C0F0          B + 1 = B              % INCREMENT RA                    02961000 D
086E  51C0 00C0 0000 C060          REGSTACK - 1 = CPCR    % COMPUTE NEXT RA INTO MAR2      02962000 D
086F  4809 0F41 0020 00F0          BMAR L = BR1           % RA + 1 = BR1                    02963000 D
0870  0000 0000 0030 0030          COMP B = SAR                                             02964000 D
0871  4809 E0C0 1000 00F0          A3 + 1 = A3            % BR2 + 1 = BR2; INCREMENT Y      02965000 D
0872  4809 E001 0010 00FC          A3 L = BR2                                               02966000 D
0873  4809 C0DE 2000 00FC          A2 - 1 = A2            % DECREMENT COUNTER               02967000 D
0874  4809 C012 0030 00FC          A2 EQL 0                                                 02968000 D
0875  6819 0000 0000 00F0          IF TRUE THEN SKIP                                        02969000 D
0876  3660 00F0 0030 C040          LDMUL - 1 = MPCR                                         02970000 D
0877  66EC 0000 0030 0040          OPCODE - 1 = MPCR                                        02971000 D
                                                          %                                 02972000 D
                                                          %                                 02973000 D
                           OPCODE04:                      % THIS ROUTINE ANALYZES THE 04 OPCODE 02974000 D
0878  5F90 0000 0030 006C          XFCODE - 1 = CPCR      % :F: FIELD IN A2                 02975000 D
0879  4809 C640 0030 00F0          A2 + AMPCR = AMPCR                                       02976000 D
087A  1250 C000 0090 00C0          OP04F - 1 = AMPCR                                        02977000 D
087B  4809 C0C0 0090 00F0          STEP                                                     02978000 D
087C  4824 00C0 C090 00F0          EXEC                                                     02979000 D
                                                          %                                 02980000 D
087D  1260 0000 0030 C040  OP04F:  OP040 - 1 = MPCR                                         02981000 D
087E  3000 C000 0000 C040          FAULT - 1 = MPCR                                         02982000 D
087F  3000 00C0 C00F 0040          FAULT - 1 = MPCR                                         02983000 D
0880  1270 0000 0030 C040          OP043 - 1 = MPCR                                         02984000 D
                                                          %                                 02985000 D
                           OP040:                         % THIS ROUTINE ISOLATES THE :M: FIELD 02986000 D
0881  4809 2C56 2000 00FC          LIT AND B = A2                                           02987000 D
0882  00FC 0000 0000 00E0          15 = LIT                                                 02988000 D
0883  4809 C15E 0030 00F0          A2 GEQ LIT                                               02989000 D
0884  0040 00C0 C000 C0E0          Q = LIT                                                  02990000 D
0885  780B 00C0 0030 00FC          IF TRUE THEN STEP ELSE SKIP                              02991000 D
0886  3000 0000 C00C C040          FAULT - 1 = MPCR                                         02992000 D
0887  4809 C640 0090 C0F0          A2 + AMPCR = AMPCR                                       02993000 D
0888  128C 00C0 0C00 00C0          OP04M - 1 = AMPCR                                        02994000 D
0889  4809 0C00 000C 00F0          STEP                                                     02995000 D
088A  4824 00C0 000C 00F0          EXEC                                                     02996000 D
                                                          %                                 02997000 D
```

```
088C  12A0 0003 0000 C040          OP0401 - 1 = MPCR                                  02999C00 D
088D  12B0 0000 0000 004C          OP04C2 - 1 = MPCR                                  03000C00 D
088E  12C0 0003 0000 C040          OP0403 - 1 = MPCR                                  03001C00 D
                                                                  %                   03002C00 D
                                   OP04C0:                        % SQUARE ROOT        03003C00 D
                                                                  % SQRT(R(A),R(A+1)) = R(A+1); REM=R(A)  03004C00 D
                                                                  % THIS ROUTINE CALCULATES THE SQUARE ROOT03005C00 D
                                                                  % OF A NUMBER PASSED VIA THE B REGISTER. 03006C00 D
                                                                  % THE SQUARE ROOT IS RETURNED IN A3,    03007C00 D
                                                                  % REMAINDER IN A2.                      03008C00 D
                                                                  % SET OV AND CC BITS                    03009C00 D
088F  4809 0C40 8B30 00FC          B R = B                                            03010C00 D
0890  80FC 0000 0000 008C          15 = LIT; 4 = SAR                                  03011C00 D
0891  48C9 2C56 0B00 00F0          LIT AND B = B                  % ISOLATE "A" FIELD  03012C00 D
0892  51C0 0000 0000 0060          REGSTACK - 1 = CPCR            % R(A) INTO MAR2    03013C00 D
0893  2F30 0000 0000 006C          EINPUT - 1 = CPCR              % (R(A)) INTO B     03014C00 D
0894  4809 0C41 2030 00FC          B L = A2                                           03015C00 D
0895  0000 0000 0000 0020          COMP 16 = SAR                                      03016C00 D
0896  4809 0F47 0020 00F0          BMAR + 1 L = BR1               % SAVE R(A+1) IN BR1 03017C00 D
0897  0000 0000 009C 030C          COMP 8 = SAR                                       03018C00 D
0898  48C9 0F46 0B30 00FC          BMAR + 1 = B                   % R(A+1)            03019C00 D
0899  51C0 0000 0000 0060          REGSTACK - 1 = CPCR            % R(A+1) INTO MAR2  03020C00 D
089A  2F30 0000 0000 006C          EINPUT - 1 = CPCR              % (R(A+1)) INTO B   03021C00 D
089B  4809 CC5C 0B30 00FC          A2 OR B = B                    % B = (R(A),R(A+1))  03022C00 D
089C  4809 CC40 0C91 00F0          B = MIR, LCTR                  % MIR USED AS TEMP STORAGE 03023C00 D
089D  A0F0 0000 0000 00B0          30 = SAR; 15 = LIT             % CTR SET FOR 16 ITERATIONS 03024C00 D
089E  4809 0003 0010 00F0          0 = BR2                                            03025C00 D
089F  4809 C003 2C1C 00F0          0 = A2,MAR2                                        03026C00 D
08A0  28C9 0003 0000 00F0          PAIR:  BMI; IF LC1                                 03027C00 D
08A1  4809 0C40 9002 C0F0          B R = A3, INC                  % FETCH NEXT MS PAIR OF DIGITS 03028C00 D
08A2  A000 0000 0000 0030          30 = SAR                                           03029C00 D
08A3  4809 EF5C 1030 00F0          A3 OR BMAR = A3                % A3 = PREV. REM/NEXT PAIR OF DIGITS 03030C00 D
08A4  4809 0C41 0030 00F0          B L = MIR                      % TEMP STORAGE       03031C00 D
08A5  A000 0000 0000 0030          COMP 2 = SAR                   % SHIFT NEXT PAIR OF DIGITS INTO MS BITS 03032C00 D
08A6  8A4B 00C3 0C00 00FC          IF COV THEN SET LC2; STEP ELSE SKIP                03033C00 D
08A7  12D0 0000 0C00 004C          SEND - 1 = MPCR                % FINISHED           03034C00 D
08A8  3000 0000 0000 C030          COMP 1 = SAR                                       03035C00 D
08A9  4809 E012 0030 00F0          A3 EQL 0                       % CHECK FOR 00 PAIR OF DIGITS 03036C00 D
08AA  6C0B C001 2030 C0FC          IF TRUE THEN A2 L = A2; STEP ELSE SKIP             03037C00 D
08AB  89FC 0000 0000 C040          PAIR - 1 = MPCR                % EXAMINE NEXT PAIR OF DIGITS 03038C00 D
08AC  4809 C001 0B30 C0F0          SEND:  A2 L = B                % PARTIAL SORT EFFECTIVELY MULT BY 4 03039C00 D
08AD  3000 0000 0030 0030          COMP 1 = SAR                                       03040C00 D
08AE  4809 0C46 0B00 00FC          B + 1 = B                                          03041C00 D
08AF  48C9 EC5F 0C1C C0F0          A3 - B L = MAR2                % POTENTIAL REMAINDER 03042C00 D
08B0  A000 0000 0000 0030          COMP 2 = SAR                                       03043C00 D
08B1  4F09 EC01 001C C0FC          IF MST THEN A3 L = MAR2; SET LC1; SKIP             03044C00 D
08B2  4809 C00C 2000 C0FC          A2 OR 1 = A2                   % BUILD PARTIAL SORT 03045C00 D
08B3  4809 C001 2000 0CF0          A2 L = A2                      % PREPARE FOR NEXT DIGIT 03046C00 D
08B4  0000 0000 0030 0030          COMP 1 = SAR                                       03047C00 D
08B5  3006 0000 0030 00F0          IF NOT LC2 THEN STEP ELSE SKIP                     03048C00 D
08B6  89FC 0000 0000 C040          PAIR - 1 = MPCR                                    03049C00 D
08B7  2C19 E000 0B00 00F0          IF LC1 THEN A3 = B; SKIP  % RESTORE REMAINDER INTO B 03050C00 D
08B8  4809 0F40 8B00 00FC          BMAR R = B                     % RESTORE REMAINDER INTO B 03051C00 D
08B9  2000 C000 0000 0000          2 = SAR                                            03052C00 D
08BA  4809 C000 9C00 00F0          A2 R = A3                      % REQUIRED TO BALANCE LAST UNUSED L SHIFT 03053C00 D
08BB  1000 0000 0030 0000          1 = SAR                                            03054C00 D
08BC  4809 0C40 2000 00F0          B = A2                         % REMAINDER INTO A2  03055C00 D
08BD  4809 E000 0B30 00FC          A3 = MIR,B                                         03056C00 D
08BE  2000 0000 0000 006C          SETCCA - 1 = CPCR              % SET THE CONDITION BITS 03057C00 D
```

```
08BF  4809 0F40 0000 2080  NOP
08C0  4809 0F40 801C 00F0  BMAR R = MAR2          % (RA+1)                    03059000 D
08C1  0000 0000 0030 C010  B = SAR                                             03060000 D
08C2  2F50 0000 0000 0060  EOUTPUT - 1 = CPCR     % SQRT INTO R(A+1)           03061000 D
08C3  4809 0F40 1030 00F0  BMAR = A3                                           03062000 D
08C4  4809 E00E 0C1C 00F0  A3 - 1 = MAR2          % R(A)                       03063000 D
08C5  4809 C000 0050 00F0  A2 = MIR                                            03064000 D
08C6  2F50 0000 0000 0060  EOUTPUT - 1 = CPCR     % REMAINDER INTO R(A)        03065000 D
08C7  5050 0000 0000 C060  CLEAROV - 1 = CPCR                                  03066000 D
08C8  66E0 0000 0030 004C  OPCODE - 1 = MPCR                                   03067000 D
                                                  %                            03068000 D
                                                  %                            03069000 D
                                  OP0401:         % RVR; REVERSE REGISTER. (REVERSE RA)  03070000 D
08C9  4809 0C40 8B00 C0F0  B R = B                % :A: FIELD LS BITS OF B      03071000 D
08CA  80FC 0000 0030 0080  4 = SAR; 15 = LIT                                   03072000 D
08CB  4809 2C56 0B00 00F0  LIT AND 6 = B          % B = :A: FIELD              03073000 D
08CC  51C0 0000 0070 0060  REGSTACK - 1 = CPCR    % RA = MAR2                  03074000 D
08CD  2F30 0000 0000 0060  EINPUT - 1 = CPCR      % B = (R(A))                 03075000 D
08CE  4809 0C40 1000 00F0  B = A3                                              03076000 D
08CF  4809 0000 2000 C0F0  0 = A2                 % ZERO A2                    03077000 D
0800  00E0 0000 0000 C0E0  14 = LIT                                            03078000 D
0801  4812 0000 0001 C0F0  LCTR; SAVE             % EXECUTE LOOP 16 TIMES      03079000 D
0802  4809 E000 0000 C0F0  A3                                                  03080000 D
0803  5C19 C000 2000 00F0  IF LST THEN A2 OR 1 = A2; SKIP  % WRITE 1 INTO LSB OF A2  03081000 D
0804  4809 C00E 2000 C0F0  A2 AND B110 = A2       % WRITE 0 INTO LSB OF A2     03082000 D
0805  4809 E000 9000 40F0  A3 R = A3,CSAR         %  MOVE (R(A)) RIGHT 1 BIT   03083000 D
0806  1000 0000 0000 0000  1 = SAR                % BIT SHIFT VALUE             03084000 D
0807  8429 C001 2002 C0F0  IF NOT COV THEN A2 L = A2,INC; JUMP                  03085000 D
0808  4809 C000 0B50 C0F0  A2 = B,MIR             % A2 CONTAINS REVERSE REG.    03086000 D
0809  2000 0000 0000 0060  SETCCA - 1 = CPCR                                   03087000 D
080A  2F50 0000 0000 0060  EOUTPUT - 1 = CPCR                                  03088000 D
080B  66E0 0000 0000 0040  OPCODE - 1 = MPCR                                   03089000 D
                                                  %                            03090000 D
                                  OP0402:         % CNT; COUNT ONES; COUNT -> RA + 1  03091000 D
080C  4809 0C40 8B00 00F0  B R = B                % :A: FIELD LS BITS OF B      03092000 D
080D  80F0 0000 0030 0080  4 = SAR; 15 = LIT                                   03093000 D
080E  4809 2C56 0B30 00F0  LIT AND B = B                                       03094000 D
080F  51C0 0000 0000 0060  REGSTACK - 1 = CPCR    % RA=MAR2                    03095000 D
08E0  2F30 0000 0000 0060  EINPUT - 1 = CPCR      % B = (R(A))                 03096000 D
08E1  4809 C000 2090 C0F0  0 = A2,MIR             % ZERO COUNT REGISTER         03097000 D
08E2  00E0 0000 0000 00E0  14 = LIT                                            03098000 D
08E3  4812 0000 0001 00F0  LCTR; SAVE             % EXECUTE LOOP 16 TIMES      03099000 D
08E4  4809 0C40 0030 00F0  B                                                   03100000 D
08E5  5C09 C000 2090 00F0  IF LST THEN A2 + 1 = A2,MIR  % INCREMENT COUNTER    03101000 D
08E6  4809 0C40 8B00 00F0  B R = B                                             03102000 D
08E7  1000 0000 0030 C000  1 = SAR                % SHIFT RA RIGHT              03103000 D
08E8  8429 0000 0072 00F0  IF NOT COV THEN INC; JUMP                           03104000 D
08E9  4809 0F46 0B30 00F0  BMAR + 1 = B           % R(A+1)                     03105000 D
08EA  51C0 0000 0000 0060  REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2           03106000 D
08EB  2F50 0000 0000 0060  EOUTPUT - 1 = CPCR     % STORE COUNT IN MIR INTO RA + 1  03107000 D
08EC  66E0 0000 0030 0040  OPCODE - 1 = MPCR                                   03108000 D
                                                  %                            03109000 D
                                  OP0403:         % SFR; SCALE FACTOR. (RA,RA+1) SHIFTED  03110000 D
                                                  % LEFT UNTIL (R(A))15 NEQ (R(A))14  03111000 D
                                                  % SHIFT COUNT -> RA+2         03112000 D
08ED  4809 0C40 8B00 00F0  B R = B                                             03113000 D
08EE  80F0 0000 0000 C080  4 = SAR; 15 = LIT                                   03114000 D
08EF  4809 2C56 0B00 C0F0  LIT AND B = B          % :A: FIELD INTO B           03115000 D
08F0  51C0 0000 0030 0060  REGSTACK - 1 = CPCR    % RA = MAR2                  03116000 D
08F1  2F30 0000 0000 0060  EINPUT - 1 = CPCR      % B = (R(A))                 03117000 D
```

```
08F2  4809 0C41 2C00 24F0          B L = A2; ASE                                  03118C00 D
08F3  0000 C000 0000 0020          COMP 16 = SAR                                  03119C00 D
08F4  4809 0F41 0C20 00F0          BMAR L = BR1            % STORE RA INTO BR1    03120C00 D
08F5  0000 C000 0000 0030          COMP 8 = SAR                                   03121000 D
08F6  4809 0F46 C000 00F0          BMAR + 1 = B            % R(A+1)               03122C00 D
08F7  51C0 0000 0000 0060          REGSTACK - 1 = CPCR     % R(A+1) INTO MAR2     03123C00 D
08F8  2F30 00C0 0030 0060          EINPUT - 1 = CPCR       % B = (R(A+1))         03124C00 D
08F9  4809 20C0 C030 C0FC          LIT = MIR                                      03125C00 D
08FA  01F0 0000 0000 C0E0          31 = LIT                                       03126C00 D
08FB  4809 CC5C 2C00 24F0          A2 OR B = A2; ASE       % A2 = (RA,RA+1)       03127C00 D
08FC  4809 C0C0 0000 00F0          A2                                             03128C00 D
08FD  6419 C012 0030 00F0          IF NOT ABT THEN A2 EQL 0 ; SKIP                03129000 D
08FE  12E0 00C0 0000 0040          SFTSTP - 1 = MPCR       % STOP COUNT; COUNT = 31 03130C00 D
08FF  680B 0000 0C00 00F0          IF TRUE THEN STEP ELSE SKIP                    03131000 D
0900  12F0 0000 0030 004C          SFTSTP - 1 = MPCR       % STOP COUNT; COUNT = 31 03132000 D
0901  4809 18C1 8B00 00F0          B101 C = B              % MASK OF TWO BITS IN B 03133C00 D
0902  11E0 0000 0000 0080          1 = SAR; 30 = LIT                              03134C00 D
0903  4809 0C00 C09C 00FC          C = MAR2,MIR                                   03135C00 D
0904  4812 00C0 0031 00F0          LCTR; SAVE                                     03136C00 D
0905  4809 CC56 9000 00FC          A2 AND B R = A3                                03137C00 D
0906  A03C 0000 0030 0080          30 = SAR; 3 = LIT                              03138C00 D
0907  4809 E012 0000 C0F0          A3 EQL 0                % TEST IF BOTH BITS EQL 0 03139C00 D
0908  6419 E152 0030 C0F0          IF FALSE THEN A3 EQL LIT; SKIP  % TEST IF BOTH BITS EQL 1 03140C00 D
0909  130C 00C0 0030 0040          SFTCNT - 1 = MPCR       % SHIFT COUNT INCREMENTED 03141C00 D
090A  6819 0000 0030 00FC          IF TRUE THEN SKIP                              03142C00 D
090B  1310 00C0 0030 C040          SFTSTP - 1 = MPCR       % STOP COUNT;  MIR AND B HAVE COUNT 03143C00 D
090C  4809 C0C1 2000 00F0  SFTCNT: A2 L = A2               % SHIFT (RA,RA+1) LEFT ONE POSITION 03144C00 D
090D  800C 0C00 0000 003C          COMP 1 = SAR                                   03145C00 D
090E  4809 0F45 009C C0FC          BMAR + 1 = MAR2,MIR                            03146C00 D
090F  8429 0000 0032 00F0          IF NOT COV THEN INC; JUMP                      03147C00 D
0910  4809 00C0 003C 2CF0  SFTSTP: ASR                                            03148C00 D
0911  4809 0F40 9000 00F0          BMAR R = A3             % BR1 INTO A3= RA       03149C00 D
0912  0C20 0000 0090 0090          8 = SAR; 2 = LIT                               03150C00 D
0913  4809 E140 C800 00F0          A3 + LIT = B            % B = R(A+2)           03151C00 D
0914  51C0 00C0 C030 0060          REGSTACK - 1 = CPCR                            03152C00 D
0915  2F5C 0000 C030 C060          EOUTPUT - 1 = CPCR      % COUNT INTO RA+2      03153C00 D
0916  66E0 00C0 0030 C040          OPCODE - 1 = MPCR                              03154C00 D
                                                           %                      03155C00 D
                           OP043:                          % BLX; BYTE LOAD AND INDEX BY 1 03156C00 D
0917  4849 000D 0C30 00F0          SET LC2                 % SET BYTE FLAG         03157C00 D
0918  5700 00C0 0C30 C060          RXMFIELD - 1 = CPCR     % B = Y; LC2 INDICATES BYTE 03158C00 D
0919  4809 0C41 0010 C0F0          B L = BR2                                      03159C00 D
091A  0000 00C0 0000 C03C          COMP 8 = SAR                                   03160C00 D
091B  50E0 00C0 0030 C060          EMULIN - 1 = CPCR       % B = (Y)              03161C00 D
091C  0FFC 0000 0000 0090          8 = SAR; 255 = LIT                             03162C00 D
091D  3C19 2C56 0090 00FC          IF LC2 THEN LIT AND B = MIR; SKIP  % LS BYTE OF Y 03163C00 D
091E  4809 0C40 8090 C0F0          B R = MIR               % MS BYTE OF Y          03164C00 D
091F  1C80 C000 0000 0060          AEQM - 1 = CPCR         % TEST IF A EQUAL M; B="A", A2="M" 03165C00 D
0920  51C0 00C0 0030 0060          REGSTACK - 1 = CPCR                            03166C00 D
0921  4809 C0C1 0030 00FC          BMI                                            03167C00 D
0922  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR      % BYTE LOAD INTO RA    03168C00 D
0923  2000 00C0 0030 C06C          SETCCA - 1 = CPCR                              03169C00 D
0924  380B 0000 0000 C0F0          IF LC2 THEN STEP ELSE SKIP  % LC2 SET BY AEQM ROUTINE 03170C00 D
0925  66E0 0000 C030 0040          OPCODE - 1 = MPCR                              03171C00 D
0926  4809 C0C0 0B30 00F0          A2 = B                                         03172C00 D
0927  51C0 00C0 0000 C06C          REGSTACK - 1 = CPCR     % RM = MAR2            03173C00 D
0928  2F30 0000 000C 0060          EINPUT - 1 = CPCR       % B = (R(M))           03174C00 D
0929  4809 0C46 009C 00FC          B + 1 = MIR             % MIR = (R(M)) + 1      03175C00 D
092A  2F50 00C0 0030 0060          EOUTPUT - 1 = CPCR                             03176C00 D
092B  66E0 00C0 000C 0060          OPCODE - 1 = CPCR                              03177C00 D
```

```
                                                  %                                                03179000 D
                        OPCODE05:                 % THIS ROUTINE ANALYZES THE 05 OPCODE           03180000 D
092C  5F90 0000 0000 0060       XFCODE - 1 = CPCR                                                03181000 D
092D  4809 C640 0080 C0FC       A2 + AMPCR = AMPCR                                               03182000 D
092E  1320 0000 0030 00C0       OP05F - 1 = AMPCR                                                03183000 D
092F  4809 0000 0000 C0F0       STEP                                                             03184000 D
0930  4824 0000 0000 C0FC       EXEC                                                             03185000 D
                                                  %                                                03186000 D
0931  133C 0000 0000 0040       OP05F:  .OP050 - 1 = MPCR                                        03187000 D
0932  1340 0000 0030 C040       OP051 - 1 = MPCR                                                 03188000 D
0933  3000 0000 003C C040       FAULT - 1 = MPCR                                                 03189000 D
0934  135C 0000 0000 C040       OP053 - 1 = MPCR                                                 03190000 D
                                                  %                                                03191000 D
                        OP050:                    % SBR; SET BIT 1 -> (R(A))F                     03192000 D
0935  4809 0C40 8800 00F0       B R = B                                                          03193000 D
0936  80F0 0000 0000 0080       4 = SAR; 15 = LIT         % B = :A: FIELD                         03194000 D
0937  4809 2C55 0830 C0F0       LIT AND B = B                                                    03195000 D
0938  51C0 0000 0030 C060       REGSTACK - 1 = CPCR       % MAR2 = RA                            03196000 D
0939  2F30 0000 0030 C060       EINPUT - 1 = CPCR         % B = (R(A))                            03197000 D
093A  4809 0C40 2000 00F0       B = A2                                                           03198000 D
093B  4809 E156 0800 00F0       A3 AND LIT = B            % B = :M: FIELD                         03199000 D
093C  00F0 0000 0000 00E0       15 = LIT                                                         03200000 D
093D  4809 0C55 0000 80F0       - B = SAR                 % VARIABLE SHIFT AMOUNT FOR LEFT SHIFT  03201000 D
093E  4809 00C1 0830 00F0       0001 L = B                                                       03202000 D
093F  4809 CC5C 0830 C0F0       A2 OR B = MIR,B           % SET 1 INTO (R(A))M AND STORE INTO MIR 03203000 D
0940  2F50 0000 0000 C060       EOUTPUT - 1 = CPCR        % MODIFIED (R(A)) -> RA                03204000 D
0941  2000 0000 0030 0060       SETCCA - 1 = CPCR                                                03205000 D
0942  66E0 0000 0000 0040       OPCODE - 1 = MPCR                                                03206000 D
                                                  %                                                03207000 D
                        OP051:                    % LXI; LOAD AND INDEX BY 1;                     03208000 D
                                                  % TYPE RI(2)                                    03209000 D
0943  4809 2C56 0830 00F0       LIT AND B = B                                                    03210000 D
0944  00F0 0000 0000 C0EC       15 = LIT                  % B = :M: FIELD                         03211000 D
0945  51C0 0000 0030 0060       REGSTACK - 1 = CPCR       % MAR2 = RM                            03212000 D
0946  2F30 0000 0000 0060       EINPUT - 1 = CPCR         % (R(M)) = Y* = B                       03213000 D
                                                  %                                                03214000 D
0947  4809 0C41 CC10 C0F0       LDINX1: B L = BR2                                                03215000 D
0948  0000 0000 0000 C030       COMP B = SAR                                                     03216000 D
0949  60E0 0000 0000 0060       EMULIN - 1 = CPCR         % B = (Y*) OR (Y)                      03217000 D
094A  4809 0C40 0030 00F0       B = MIR                   % MIR = (Y*) OR (Y)                     03218000 D
094B  1C80 0000 0000 006C       AEQM - 1 = CPCR           % IS A EQUAL M ; B="A", A2 = "M"        03219000 D
094C  51C0 0000 0000 0060       REGSTACK - 1 = CPCR       % RA = MAR2                             03220000 D
094D  2F50 0000 0000 0060       EOUTPUT - 1 = CPCR        % (Y) OR (Y*) -> RA                     03221000 D
094E  4809 0000 0000 C0F0       BMI                                                              03222000 D
094F  2000 0000 0000 0060       SETCCA - 1 = CPCR         % (Y) OR (Y*) -> B                      03223000 D
0950  380B 0000 0030 00FC       IF LC2 THEN STEP ELSE SKIP  % LC2 SET IN AEQM                      03224000 D
0951  56E0 0000 0030 0040       OPCODE - 1 = MPCR                                                03225000 D
0952  4809 C000 C800 C0F0       A2 = B                                                           03226000 D
0953  51C0 0000 0000 0060       REGSTACK - 1 = CPCR                                              03227000 D
0954  2F30 0000 0000 0060       EINPUT - 1 = CPCR         % B = (R(M))                           03228000 D
0955  4809 0C46 0030 00F0       B + 1 = MIR               % MIR = (R(M)) + 1                      03229000 D
0956  2F50 0000 0000 0060       EOUTPUT - 1 = CPCR                                               03230000 D
0957  56E0 0000 0000 0040       OPCODE - 1 = MPCR                                                03231000 D
                                                  %                                                03232000 D
                        OP053:                    % LXI LOAD AND INDEX BY 1                       03233000 D
                                                  % TYPE RX                                      03234000 D
0958  5700 0000 0000 C060       RXMFIELD - 1 = CPCR       % B = Y                                03235000 D
0959  946C 0000 0000 0040       LDINX1 - 1 = MPCR         % LOAD AND INDEX BY 1                   03236000 D
                                                  %                                                03237000 D
```

```
                                OPCODE06:                   % THIS ROUTINE ANALYZES THE 06 OPCODE          03238000 D
095A  5F90 0000 0030 0060                XFCODE - 1 = CPCR                                                 03239000 D
095B  4809 C640 0090 00F0                A2 + AMPCR = AMPCR                                                03240000 D
095C  1360 0000 0030 00C0                OP06F - 1 = AMPCR                                                 03241000 D
095D  4809 0000 0000 00F0                STEP                                                              03242000 D
095E  4824 0000 0030 00F0                EXEC                                                              03243000 D
                                                            %                                              03244000 D
095F  137C 0000 0030 C040       OP06F:   OP060 - 1 = MPCR                                                 03245000 D
0960  1380 0000 0000 C040                OP061 - 1 = MPCR                                                  03246000 D
0961  3000 0000 0030 C040                FAULT - 1 = MPCR                                                  03247000 D
0962  139C 0000 0030 0040                OP063 - 1 = MPCR                                                  03248000 D
                                                            %                                              03249000 D
                                OP060:                      % ZBR; ZERO BIT; C -> (R(A))M                  03250000 D
0963  4809 0C40 8B30 00F0                B R = B                                                           03251000 D
0964  80F0 0000 0030 C080                4=SAR; 15=LIT                                                     03252000 D
0965  4809 2C56 0B00 00F0                LIT AND B = B              % B = :A: FIELD                         03253000 D
0966  51C0 0000 0030 0060                REGSTACK - 1 = CPCR                                               03254000 D
0967  2F30 0000 0000 006C                EINPUT - 1 = CPCR          % B = (R(A))                            03255000 D
0968  4809 0C40 2000 C0F0                B = A2                                                            03256000 D
0969  4809 E156 0B30 00F0                A3 AND LIT = B             % B = :M: FIELD                         03257000 D
096A  0CFC 0000 0030 00E0                15 = LIT                                                          03258000 D
096B  4809 2C5E 0030 80F0                LIT - B = SAR              % VARIABLE SHIFT AMOUNT FOR RIGHT SHIFT 03259000 D
096C  0200 00C0 0030 00E0                32 = LIT                                                          03260000 D
096D  4809 0009 8B0C 00F0                B110 C = B                 % MASK                                 03261000 D
096E  4809 CC56 0B30 00F0                A2 AND B = MIR,B           % C -> (R(A))M                          03262000 D
096F  2F50 0000 0030 C06C                EOUTPUT - 1 = CPCR         % MODIFIED (R(A)) -> RA                 03263000 D
0970  66E0 0000 0000 0040                OPCODE - 1 = MPCR                                                 03264000 D
                                                            %                                              03265000 D
                                OP061:                      % LDXI; LOAD DOUBLE (INDIRECT)                  03266000 D
                                                            % INDEX BY TWO; RI(2)                          03267000 D
0971  4809 2C56 0B30 00F0                LIT AND B = B                                                     03268000 D
0972  00F0 0000 0030 00E0                15 = LIT                                                          03269000 D
0973  51C0 0000 0090 0060                REGSTACK - 1 = CPCR                                               03270000 D
0974  2F30 0000 0030 0060                EINPUT - 1 = CPCR          % B = (R(M)) = YW                       03271000 D
0975  482C 0000 0030 0060                LDBLE - 1 = CPCR                                                  03272000 D
                                                            %                                              03273000 D
0976  1CB0 0000 0030 006C       LDINX2:  AEQM - 1 = CPCR                                                   03274000 D
0977  4809 C000 0B30 00FC                A2 = B                                                            03275000 D
0978  3806 0000 0000 00F0                IF LC2 THEN STEP ELSE SKIP  % :A: = :M:                            03276000 D
0979  66E0 0000 0030 C040                OPCODE - 1 = MPCR                                                 03277000 D
097A  51C0 0000 0030 C060                REGSTACK - 1 = CPCR        % RM = MAR2                            03278000 D
097B  2F30 0000 0000 0060                EINPUT - 1 = CPCR          % B = (R(M))                            03279000 D
097C  4809 2C40 C030 C0F0                LIT + B = MIR                                                     03280000 D
097D  0020 00C0 0000 00E0                2 = LIT                                                           03281000 D
097E  2F50 0000 0000 0060                EOUTPUT - 1 = CPCR         % (R(M)) + 2 -> RM                     03282000 D
097F  66E0 0000 0000 0060                OPCODE - 1 = CPCR                                                 03283000 D
                                                            %                                              03284000 D
                                OP063:                      % LDX; LOAD DOUBLE, INDEX BY TWO                03285000 D
                                                            % TYPE RX                                     03286000 D
0980  5700 0000 0030 C060                RXMFIELD - 1 = CPCR        % B = Y                                 03287000 D
0981  482C 0000 0030 0060                LDBLE - 1 = CPCR           % LOAD DOUBLE                           03288000 D
0982  9750 0000 0030 0040                LDINX2 - 1 = MPCR          % INDEX BY TWO                          03289000 D
                                                            %                                              03290000 D
                                                            %                                              03291000 D
                                OPCODE07:                   % THIS ROUTINE ANALYZES THE 07 OPCODE           03292000 D
0983  5F90 0000 0030 C060                XFCODE - 1 = CPCR                                                 03293000 D
0984  4809 C640 0090 00F0                A2 + AMPCR = AMPCR                                                03294000 D
0985  13A0 0000 0030 00C0                OP07F - 1 = AMPCR                                                 03295000 D
0986  4809 0000 0030 C0F0                STEP                                                              03296000 D
0987  4824 0000 0030 00F0                EXEC                                                              03297000 D
```

```
098B  1380 0000 0030 0040        OP07F:  OP070 - 1 = MPCR                                    03299000 D
098C  1300 0000 0000 0040                OP071 - 1 = MPCR                                    03300000 D
098A  3000 0000 0030 0040                FAULT - 1 = MPCR                                    03301000 D
098B  1300 0000 0000 0040                OP073 - 1 = MPCR                                    03302000 D
                                                                  %                          03303000 D
                                 OP070:                           % CBR: COMPARE BIT: TEST BIT M OF RA FOR 03304000 D
                                                                  % ZERO                     03305000 D
098C  4809 0C40 8B00 C0F0                B R = B                                             03306000 D
098D  80F0 C000 C000 0080                4 = SAR: 15 = LIT                                   03307000 D
098E  4809 2C55 0B70 C0F0                LIT AND B = B                                       03308000 D
098F  5100 0000 0000 C060                REGSTACK - 1 = CPCR      % RA = MAR2                03309000 C
0990  2F30 0000 0000 C060                EINPUT - 1 = CPCR        % B = (R(A))               03310000 D
0991  4809 0C40 2000 C0F0                B = A2                                              03311000 D
0992  4809 E155 0B00 C0F0                A3 AND LIT = B           % ISOLATE :M: FIELD         03312000 D
0993  00F0 0000 0600 00E0                15 = LIT                                            03313000 D
0994  4809 0C40 000C 80F0                B = SAR                  % SHIFT COMPARE BIT TO LS POSITION 03314000 D
0995  4809 C000 A000 C0F0                A2 R = A2                                           03315000 D
0996  4809 C000 0000 00F0                A2                                                  03316000 D
0997  580B 0000 0000 00F0                IF LST THEN STEP ELSE SKIP  % TEST BIT               03317000 D
0998  13E0 C000 0000 0040                TEST1 - 1 = MPCR         % TEST FOR 1 TRUE          03318000 D
0999  4809 2C52 0000 00F0                LIT EQL B                %  BIT POSITION 15?        03319000 D
099A  6C19 1BD9 8B70 00F0                IF TRUE THEN B010 C = B: SKIP  % SET 00 INTO CC BITS 03320000 D
099B  4809 1B19 8B00 00F0                B011 C = B               % SINGLE BIT SET TO 0       03321000 D
099C  0000 0000 0000 0010                B = SAR                                             03322000 D
099D  4809 AC56 4000 00F0                A1 AND B = A1            % SET 0 OR 00 INTO CC BITS  03323000 D
099E  66E0 0000 0000 C040                OPCODE - 1 = MPCR                                   03324000 D
                                                                  %                          03325000 D
099F  4809 2C52 000C 00F0        TEST1:  LIT EQL B                % BIT POSITION 15?          03326000 D
09A0  6C19 1BC1 8B00 00F0                IF TRUE THEN B101 C = B: SKIP  % SET 11 IN BOTH CC BITS 03327000 D
09A1  4809 1BC1 8B70 00F0                B100 C = B               % SET 1 INTO LS BIT OF CC   03328000 D
09A2  0000 0000 0000 0010                B = SAR                                             03329000 D
09A3  4809 AC5C 4000 00F0                A1 OR B = A1             % SET 1 OR 11 INTO CC BIT   03330000 D
09A4  66E0 0000 0000 0040                OPCODE - 1 = MPCR                                   03331000 D
                                                                  %                          03332000 D
                                 OP071:                           % LPI: LOAD PSW (INDIRECT)  03333000 D
                                                                  % (Y*,Y*+1,Y*+2) -> P,SR1,SR2 03334000 D
09A5  4809 2C56 0B00 00F0                LIT AND B = B                                       03335000 D
09A6  00F0 0000 0000 00E0                15 = LIT                                            03336000 D
09A7  5100 0000 0000 0060                REGSTACK - 1 = CPCR      % RM = MAR2                03337000 D
09A8  2F30 0000 0000 0060                EINPUT - 1 = CPCR        % B = (R(M)) = Y*           03338000 D
09A9  4809 0C40 2000 C0F0        LPSW:   B = A2                   % STORE B IN A2             03339000 D
09AA  4809 0C41 0010 00F0                B L = BR2                % Y* = BR2 OR Y = BR2        03340000 D
09AB  000C 0000 0C0C C030                COMP 8 = SAR                                        03341000 D
09AC  60E0 0000 0000 0060                EMULIN - 1 = CPCR        % B = (Y*) OR (Y)           03342000 D
09AD  4809 A000 C000 00F0                A1 R = A1                                           03343000 D
09AE  0000 0000 0000 0020                16 = SAR                                            03344000 D
09AF  4809 A001 4000 00F0                A1 L = A1                                           03345000 D
09B0  4809 AC5C 4000 00F0                A1 OR B = A1             % PAR= (Y) OR (Y*)          03346000 D
09B1  4809 C000 2000 00F0                A2 + 1 = A2              % NEXT SEQ Y OR Y*          03347000 D
09B2  4809 C001 0010 C0F0                A2 L = BR2                                          03348000 D
09B3  0000 C000 0C0C C030                COMP 8 = SAR                                        03349000 D
09B4  60E0 0000 0000 0060                EMULIN - 1 = CPCR        % (Y*+1) OR (Y+1) = B       03350000 D
09B5  4809 0C41 0090 00F0                B L = MIR                % STORE B INTO UHW OF MIR   03351000 D
09B6  0000 C000 000C 0020                COMP 16 = SAR                                       03352000 D
09B7  4809 2001 1000 00F0                LIT L = A3               % CREATE BIT MASK FOR LOADER TOGGLES 03353000 D
09B8  3070 0000 000C 0080                7 = LIT: COMP 29 = SAR                              03354000 D
09B9  4809 2001 0B0C C0F0                LIT L = B                % STORE MASK FOR LOWER SR1 INTO B 03355000 D
09BA  0700 0000 0000 00A0                112 = LIT: COMP 16 = SAR                            03356000 D
09BB  4809 EC5C 0B00 00F0                A3 OR B = B              % CREATE FULL WORD MASK OF 1:S 03357000 D
```

```
09BC  4809 0C42 1000 00F0          NOT B = A3, BMI        % MASK FOR (RA)                  03358000 D
09BD  4809 EC56 0030 00F0          A3 AND B = MIR         % MASK OFF (RA), STORE IN MIR     03359000 D
09BE  4809 E002 0B30 00F0          NOT A3 = B             % MASK FOR A1 (PSW)              03360000 D
09BF  4809 AC56 1000 00FC          A1 AND B = A3, BMI     % MASK OFF SAVED PART OF A1 INTO A2 03361000 D
09C0  4809 EC5C 0B70 C0F0          A3 OR B = B            % CREATE NEW A1 VALUE FROM (R(A))  03362000 D
09C1  4809 A001 4000 00F0         A1 L = A1              % CLEAR UHW OF A1                  03363000 D
09C2  0000 CCC0 0030 0020          COMP 16 = SAR                                            03364000 D
09C3  4809 A000 C000 C0FC          A1 R = A1              % RESTORE A1                      03365000 D
09C4  4809 AC5C 4000 00F0          A1 OR R = A1                                             03366000 D
09C5  4809 CCC0 2030 00F0          A2 + 1 = A2            % NEXT SEQ Y OR Y*                03367000 D
09C6  4809 C001 0C10 00FC          A2 L = BR2                                               03368000 D
09C7  0000 00C0 0030 C030          COMP B = SAR                                             03369000 D
09C8  60E0 0000 0C00 006C          EMULIN - 1 = CPCR      % (Y*+2) OR (Y+2) = B             03370000 D
09C9  4809 0C40 2030 00F0          B = A2                                                   03371000 D
09CA  4809 20C0 001C 00F0          LIT = MAR2             % SR2 = MAR2                      03372000 D
09CB  0220 00C0 0000 C0E0          STATUS2 = LIT                                            03373000 D
09CC  2F30 00C0 0000 0060          EINPUT - 1 = CPCR      % B = (STATUS2)                   03374000 D
09CD  4809 0C40 8000 C0F0          B R = B                % ISOLATE UPPER 16 BITS OF STATUS2 03375000 D
09CE  0000 00C0 0030 C020          16 = SAR                                                 03376000 D
09CF  4809 0C41 0800 00FC          B L = B                                                  03377000 D
09D0  4809 CC5C 0C90 00F0          A2 OR B = MIR          % CREATE NEW STATUS2              03378000 D
09D1  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                       03379000 D
09D2  66E0 00C0 C000 0040          OPCODE - 1 = MPCR                                        03380000 D
                                                          %                                 03381000 D
                                OP073:                    % LPI LOAD PSW                    03382000 D
                                                          % (Y,Y+1,Y+2) -> P,SR1,SR2        03383000 D
09D3  570C 00C0 0000 0060          RXMFIELD - 1 = CPCR    % B = Y                           03384000 D
09D4  9A80 0000 0000 0040          LPSW - 1 = MPCR                                          03385000 D
                                                          %                                 03386000 D
                                                          %                                 03387000 D
                                OPCODE10:                 % LOGICAL RIGHT SHIFT             03388000 D
09D5  5F9C 0000 0000 0060          XFCODE - 1 = CPCR      % RETURNS "F" IN A2               03389000 D
09D6  4809 C640 0030 C0F0          A2 + AMPCR = AMPCR                                       03390000 D
09D7  13F0 00C0 0C0C C0C0          OP10F - 1 = AMPCR                                        03391000 D
09D8  4809 0000 0000 C0F0          STEP                                                     03392000 D
09D9  4824 0000 0000 C0F0          EXEC                                                     03393000 D
                                                          %                                 03394000 D
                                                          %                                 03395000 D
09DA  1400 0000 C000 0040       OP10F:  OP100 - 1 = MPCR                                    03396000 D
09DB  3000 0000 0000 0040          FAULT - 1 = MPCR                                         03397000 D
09DC  1410 0000 0000 0040          OP102 - 1 = MPCR                                         03398000 D
09DD  1420 0000 0000 0040          OP103 - 1 = MPCR                                         03399000 D
                                                          %                                 03400000 D
                                                          %                                 03401000 D
                                OP100:                    % RR TYPE LOGICAL RIGHT SHIFT     03402000 D
                                                          % SHIFT (R(A)) RIGHT BY THE LS 5 BITS 03403000 D
                                                          % IN (R(M)), ZERO FILL, AND SET CC 03404000 D
09DE  2380 0000 0000 006C          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                   03405000 D
09DF  4809 0C40 0C90 00F0          E = MIR                % MIR USED AS TEMP FOR (R(M))      03406000 D
09E0  4809 E000 0B70 00F0          A3 = B                                                   03407000 D
09E1  228C 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B, R(A) IN MS WORD OF A3 03408000 D
09E2  4809 0C40 1D30 00FC          B = A3, BMI                                              03409000 D
09E3  4809 0C40 0030 80F0          B = SAR                % SAR HOLDS BITS 0-5 OF (R(M))    03410000 D
09E4  4809 E000 8880 00F0          A3 R = MIR, B                                            03411000 D
09E5  20D0 0000 0000 0060          SETCCA - 1 = CPCR      % SET CONDITION BITS              03412000 D
09E6  2F50 00C0 0000 0060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A))                03413000 D
09E7  66E0 0000 0030 C040          OPCODE - 1 = MPCR                                        03414000 D
                                                          %                                 03415000 D
                                                          %                                 03416000 D
                                OP102:                    % RK TYPE LOGICAL RIGHT SINGLE SHIFT 03417000 D
```

```
                                                  % SHIFT (R(A)) RIGHT Y BITS 0-5 PLACES   03418000 D
                                                  % ZERO FILL AND SET CC                   03419000 D
09E8  6330 0000 0030 0060        IFETCH - 1 = CPCR % Y INTO B                              03420000 D
09E9  4809 0C43 0030 C0F0        B = MIR                                                   03421000 D
09EA  4809 E156 0B30 C0F0        A3 AND LIT = B    % ISOLATE "H" FIELD                     03422000 D
09EB  00F0 0000 0000 00E0        15 = LIT                                                  03423000 D
09EC  4809 0C52 0C00 00F0        C EQL B                                                   03424000 D
09ED  6C19 0000 0B00 00F0        IF TRUE THEN 0 = B; SKIP                                  03425000 D
09EE  238C 0000 0000 006C        CONTENTSRM - 1 = CPCR  % (R(H)) INTO B                    03426000 D
09EF  4809 0C40 2000 00F0        B = A2, BMI       % TRANSFER (R(H)) INTO A2                03427000 D
09F0  4809 CC40 0C90 00F0        A2 + B = MIR      % Y INTO MIR                             03428000 D
09F1  4809 E0C0 0B0C C0F0        A3 = B            % INSTRUCTION INTO B                     03429000 D
09F2  22B0 00C0 000C 006C        CONTENTSRA - 1 = CPCR % (R(A)) INTO B                     03430000 D
09F3  4809 0C40 1D30 00F0        B = A3, BMI       % (R(A)) INTO A3                         03431000 D
09F4  4809 0C40 0030 80F0        B = SAR           % Y(0-5) INTO SAR                        03432000 D
09F5  4809 E0C0 8B30 0CF0        A3 R = MIR, B     % PERFORM SHIFT                          03433000 D
09F6  2F5C 00C0 0C00 006C        EOUTPUT - 1 = CPCR                                        03434000 D
09F7  2000 0000 0030 C060        SETCCA - 1 = CPCR % SET APPROPRIATE CONDITION BITS        03435000 D
09F8  66E0 00C0 000C 0040        OPCODE - 1 = MPCR                                         03436000 D
                                                  %                                        03437000 D
                                                  %                                        03438000 D
                         OP103:                   % RX TYPE BYTE STORE                      03439000 D
                                                  % (R(A)) BITS 0-7 INTO Y BYTE             03440000 D
09F9  4849 0000 003C C0F0        SET LC2           % LC2 USED AS BYTE OPERATION FLAG        03441000 D
09FA  5700 00C0 0C30 006C        RXMFIELD - 1 = CPCR % Y ADDR INTO B                       03442000 D
09FB  4809 0C41 0010 00F0        B L = BR2         % Y INTO BR2                             03443000 D
09FC  000C 00C3 0C30 003C        COMP 8 = SAR                                              03444000 D
09FD  4809 0F40 0020 C0F0        EMAR = BR1        % STORE Y                                03445000 D
09FE  60EC 0000 0000 006C        EMULIN - 1 = CPCR % (Y) INTO B                            03446000 D
09FF  4809 0C40 0C90 00F0        B = MIR           % (Y) INTO MIR                           03447000 D
0A00  4809 E0C0 0B0C C0F0        A3 = B            % RESTORE INSTRUCTION INTO B             03448000 D
0A01  22B0 0000 0030 C060        CONTENTSRA - 1 = CPCR % (R(A)) INTO B                     03449000 D
0A02  4809 2C57 1D0C C0F0        LIT AND B L = A3, BMI % (R(A)) BITS (0-7) INTO 2ND LSB OF A3 03450000 D
0A03  0FFC 0000 000C 008C        255 = LIT; COMP 8 = SAR % (Y) INTO B                      03451000 D
0A04  380B 00C0 0000 0CF0        IF LC2 THEN STEP ELSE SKIP % IF LC2 SET, PUT BYTE INTO    03452000 D
0A05  143C 0000 0C30 0040        LS103 - 1 = MPCR  % LS BYTE OF Y                          03453000 D
0A06  4809 0C41 0B00 40F0        B L = B, CSAR     % CLEAR THE MS BYTE OF B                03454000 D
0A07  0000 0000 0C30 0010        COMP 24 = SAR                                             03455000 D
0A08  4809 0C40 8B00 00F0        B R = B                                                   03456000 D
0A09  4809 E000 9000 40F0        A3 R = A3, CSAR   % PUT (R(A)) -BITS 0-7 INTO 2ND LS BYTE 03457000 D
0A0A  0000 00C0 0000 001C        8 = SAR           % IN A3                                 03458000 D
0A0B  4809 E0C1 1030 C0F0        A3 L = A3                                                 03459000 D
0A0C  4809 EC5C 0C90 C0F0        A3 OR B = MIR     % MIR CONTAINS NEW (Y)                  03460000 D
0A0D  1440 0000 0030 0040        CNT103 - 1 = MPCR % USED AS A "GO TO"                     03461000 D
                         LS103:                   % (R(A)) BITS 0-7 INTO LS BYTE OF A3     03462000 D
0A0E  4809 0C40 8B00 40F0        B R = B, CSAR     % CLEAR LS BYTE OF B                    03463000 D
0A0F  000C 00C0 0030 001C        8 = SAR                                                   03464000 D
0A10  4809 0C41 0B30 40F0        B L = B, CSAR                                             03465000 D
0A11  4809 E000 9030 00F0        A3 R = A3         % (R(A)) BITS 0-7 INTO LS BYTE OF A3    03466000 D
0A12  4809 EC5C 0C90 C0F0        A3 OR B = MIR     % MIR CONTAINS NEW (Y)                  03467000 D
0A13  4809 0000 0030 20F0        ASR               % REFERENCE BR1                         03468000 D
0A14  4809 0F40 0010 00F0        EMAR = BR2        % Y INTO BR2                            03469000 D
0A15  61EC 00C0 C000 0060        EMULOUT - 1 = CPCR % WRITE INTO R(A)                      03470000 D
0A16  66E0 00C0 0000 0040        OPCODE - 1 = MPCR                                         03471000 D
                                                  %                                        03472000 D
                                                  %                                        03473000 D
                         OPCODE11:                % RIGHT SHIFT AND STORE                   03474000 D
0A17  5F90 0000 003C 0060        XFCODE - 1 = CPCR % "F" INTO A2                           03475000 D
0A18  4809 C640 0030 00F0        A2 + AMPCR = AMPCR                                        03476000 D
0A19  145C 00C0 0000 C0C0        OP11F - 1 = AMPCR                                         03477000 D
```

```
0A1A  4809 0000 0030 00F0          STEP                                          03478000 D
0A1B  4824 0000 0030 00F0          EXEC                                          03479000 D
                                                          %                      03460000 D
                                                          %                      03461000 D
0A1C  1460 0000 0030 0040   OP11F: OP110 - 1 = MPCR                              03482000 D
0A1D  1470 0000 0030 0040          OP111 - 1 = MPCR                              03483000 D
0A1E  1480 0000 0030 0040          OP112 - 1 = MPCR                              03484000 D
0A1F  1490 0000 0030 0040          OP113 - 1 = MPCR                              03485000 D
                                                          %                      03486000 D
                                                          %                      03487000 D
                            OP110:                        % RR TYPE ALG RIGHT SINGLE SHIFT  03488000 D
                                                          % SHIFT R(A) RIGHT (R(M)) (0-5)  03489000 D
                                                          % SIGN FILL AND SET CC  03490000 D
0A20  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR % (R(M)) INTO B          03491000 D
0A21  4809 0C40 0030 00F0          B = MIR                % TEMP STORAGE           03492000 D
0A22  4809 E000 0B00 00F0          A3 = B                 % PUT INSTR BACK INTO B  03493000 D
0A23  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B         03494000 D
0A24  4809 0C40 1000 00F0          B = A3, BMI            % (R(A)) INTO A3, (R(M)) INTO B  03495000 D
0A25  4809 E001 9000 40F0          A3 C = A3, CSAR        % POS. SIGN BIT OF (R(A)) INTO LS BIT  03496000 D
0A26  9000 0000 0000 0010          15 = SAR                                      03497000 D
0A27  4809 E000 0000 00F0          A3                                            03498000 D
0A28  5BC9 0000 0000 C0F0          IF LST THEN SET LC1    % FLAG FOR A NEG. (R(A))  03499000 D
0A29  4809 E001 9000 00F0          A3 C = A3              % RESTORE A3            03500000 D
0A2A  4809 0C40 2000 00F0          B = A2                 % A2 CONTAINS (R(M))    03501000 D
0A2B  4809 0002 0B00 00F0          NOT 0 = B              % FILL B WITH ALL 1:S   03502000 D
0A2C  4809 0C41 0B00 00F0          B L = B                % B = 1111/0000          03503000 D
0A2D  0000 0003 0000 0020          16 = SAR                                      03504000 D
0A2E  2C09 EC5C 1000 00F0          IF LC1 THEN A3 OR B = A3                      03505000 D
0A2F  4809 C000 0030 80F0          A2 = SAR               % SAR CONTAINS (R(M)) (0-5)  03506000 D
0A30  4809 E000 8B00 00F0          A3 R = B               % PERFORM ALG SHIFT      03507000 D
0A31  4809 0C41 0B70 00F0          B L = B                % CLEAR UHW B            03508000 D
0A32  0000 0003 0030 0C20          COMP 16 = SAR                                 03509000 D
0A33  4809 0C40 8B30 00F0          B R = B, MIR           % RESTORE (R(A))         03510000 D
0A34  2F5C 0000 0030 0060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A))      03511000 D
0A35  2000 0000 0000 006C          SETCCA - 1= CPCR       % ARTHMETIC CC SETTING (ENTER WITH B)  03512000 D
0A36  66E0 0000 0000 0040          OPCODE - 1 = MPCR                             03513000 D
                                                          %                      03514000 D
                                                          %                      03515000 D
                            OP111:                        % RI TYPE 2 STORE, (R(A)) INTO Y*  03516000 D
0A37  2280 0000 0000 C060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B, R(A) INTO MS A3  03517000 D
0A38  4809 0C40 C080 00FC          B = MIR                % TEMP STORAGE OF (R(A))  03518000 D
0A39  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % Y* INTO B             03519000 D
0A3A  4809 0C41 0010 00F0          B L = BR2                                     03520000 D
0A3B  0000 0000 0030 0030          COMP 8 = SAR                                  03521000 D
0A3C  61E0 0000 0000 C060          EMULOUT - 1 = CPCR     % (R(A)) INTO Y*        03522000 D
0A3D  66E0 0000 0030 0C40          OPCODE - 1 = MPCR                             03523000 D
                                                          %                      03524000 D
                                                          %                      03525000 D
                            OP112:                        % RK TYPE, ALG, RIGHT SINGLE SHIFT  03526000 D
                                                          % SHIFT (R(A)) RIGHT Y (0-5)  03527000 D
                                                          % SIGN FILL, AND SET CC  03528000 D
0A3E  4809 2C56 0B00 00F0          LIT AND B = B          % ISOLATE "M" FIELD      03529000 D
0A3F  00F0 0000 0070 00EC          15 = LIT                                      03530000 D
0A40  4809 0C52 0030 00FC          B EQL 0                                       03531000 D
0A41  6819 0000 0000 00F0          IF TRUE THEN SKIF                             03532000 D
0A42  2380 0000 0030 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B         03533000 D
0A43  4809 0C41 0B30 00F0          B L = B                                       03534000 D
0A44  0000 0000 0030 0020          COMP 16 = SAR                                 03535000 D
0A45  4809 EC5C 1000 00F0          A3 OR B = A3           % A3 = (R(M))/INSTRUCTION  03536000 D
0A46  633C 0000 0000 0060          IFETCH - 1 = CPCR      % Y INTO B              03537000 D
```

```
0A47  4809 E0C0 AC00 C0F0          A3 R = A2                  % (R(M)) INTO A2                03538C00 D
0A48  0000 00C0 0C00 CC2C          16 = SAR                                                   03539C00 D
0A49  4809 CC40 2000 00F0          A2 + B = A2                % Y INTO A2                      03540C00 D
0A4A  4809 E0C0 9000 00F0          A3 R = A3                                                  03541C00 D
0A4B  80FC 00C0 0000 0080          4 = SAR; 15 = LIT                                          03542C00 D
0A4C  4809 E156 CB00 C0F0          A3 AND LIT = B             % ISOLATE "A" FIELD             03543C00 D
0A4D  51C0 00C0 0C00 006C          REGSTACK - 1 = CPCR        % ADDR OF P(A) INTO MAR2        03544C00 D
0A4E  2F30 00C0 0C00 0060          EINPUT - 1 = CPCR          % (R(A)) INTO B                 03545C00 D
0A4F  4809 0C40 9000 C0F0          B R = A3                   % PREPARE TO EXAMINE SIGN BIT    03546C00 D
0A50  8000 00C0 0000 CC10          15 = SAR                                                   03547C00 D
0A51  4809 E0C0 0000 00F0          A3                                                         03548C00 D
0A52  5C09 0002 1000 00F0          IF LST THEN NOT 0 = A3                                     03549C00 D
0A53  4809 E3C1 1C00 00F0          A3 L = A3                  % A3 = 1111/0000                03550C00 D
0A54  0000 0000 0C00 0020          COMP 16 = SAR                                              03551C00 D
0A55  4809 EC5C 0000 00F0          A3 OR B = B                % P = 1111/(R(A)) OR 0000/(R(A)) 03552C00 D
0A56  4809 C0C0 0000 80F0          A2 = SAR                   % Y (0-5) INTO SAR               03553C00 D
0A57  4809 0C40 8B00 00F0          B R = B                    % PERFORM ALG SHIFT              03554C00 D
0A58  4809 0C41 0B0C C0F0          B L = B                    % CLEAR UHW B                    03555C00 D
0A59  0000 0000 0C00 C020          COMP 16 = SAR                                              03556C00 D
0A5A  4809 0C40 8B80 00F0          B R = B; MIR               % RESTORE SHIFTED (R(A))         03557C00 D
0A5B  2F50 00C0 0000 0060          EOUTPUT - 1 = CPCR         % MAR2 HAS ADDR OF R(A)          03558C00 D
0A5C  2000 0000 0000 006C          SETCCA - 1 = CPCR          % SET THE CONDITION BITS         03559C00 D
0A5D  56EC 0000 0000 C060          OPCODE - 1 = CPCR                                          03560C00 D
                                                              %                                 03561C00 D
                                                              %                                 03562C00 D
                                   OP113:                     % RX TYPE STORE, (R(A)) INTO Y     03563C00 D
0A5E  5700 00C0 0C0C 0060          RXMFIELD - 1 = CPCR        % B HOLDS ADDR OF Y              03564C00 D
0A5F  4809 0C41 0030 00F0          B L = MIR                  % Y INTO MIR                     03565C00 D
0A60  0000 0000 0C00 003C          COMP 8 = SAR                                               03566C00 D
0A61  4809 E0C0 9000 00F0          A3 R = A3                  % SHIFT "A" TO LS BITS           03567C00 D
0A62  90FC 00C0 0000 0080          4 = SAR; 15 = LIT                                          03568C00 D
0A63  4809 E156 CB00 00F0          A3 AND LIT = B             % "A" INTO B                    03569C00 D
0A64  51C0 0000 0C00 C060          REGSTACK - 1 = CPCR        % P(A) INTO MAR2                03570C00 D
0A65  2F3C 0000 0C00 C06C          EINPUT - 1 = CPCR          % (R(A)) INTO B                 03571C00 D
0A66  4809 0C40 0030 00F0          B = MIR,BMI                % (R(A)) INTO MIR                03572C00 D
0A67  4809 0C43 0010 C0F0          B = BR2                    % Y INTO BR2                     03573C00 D
0A68  61EC 00C0 0C00 0060          EMULOUT - 1 = CPCR         % (R(A)) INTO Y                  03574C00 D
0A69  66EC 0000 0030 0040          OPCODE - 1 = MPCR                                          03575C00 D
                                                              %                                 03576C00 D
                                                              %                                 03577C00 D
                                   OPCODE12:                  % STORE AND RIGHT SHIFT          03578C00 D
0A6A  5F90 00C0 0000 C060          XFCODE - 1 = CPCR          % "F" IN A2                      03579C00 D
0A6B  4809 C640 0030 00F0          A2 + AMPCR = AMPCR                                         03580C00 D
0A6C  14AC 0000 0C00 00C0          OP12F - 1 = AMPCP                                          03581C00 D
0A6D  4809 0000 0C00 00F0          STEP                                                       03582C00 D
0A6E  4824 0000 0000 00F0          EXEC                                                       03583C00 D
                                                              %                                 03584C00 D
                                                              %                                 03585C00 D
0A6F  14B0 00C0 0000 0040          OP12F:  OP120 - 1 = MPCR                                   03586C00 D
0A70  14C0 0000 0C00 0040          OP121 - 1 = MPCR                                           03587C00 D
0A71  14DC 00C0 0C00 0040          OP122 - 1 = MPCR                                           03588C00 D
0A72  14EC 00C0 0030 0040          OP123 - 1 = MPCR                                           03589C00 D
                                                              %                                 03590C00 D
                                                              %                                 03591C00 D
                                   OP120:                     % RR TYPE LOGICAL RIGHT DOUBLE SHIFT 03592C00 D
                                                              % SHIFT R(A), R(A+1) RIGHT (R(M)) (0-5) 03593C00 D
                                                              % ZERO FILL AND SET CC            03594C00 D
0A73  4809 0C40 8B0C 00F0          B R = B                                                    03595C00 D
0A74  90F0 00C0 0C0C 0080          4 = SAR; 15 = LIT                                          03596C00 D
0A75  4809 2C56 0B00 00F0          LIT AND B = B              % ISOLATE "A" FIELD              03597C00 D
```

```
0A76  51C0 00C0 0000 0060          REGSTACK - 1 = CPCR      % ADDR OF R(A) INTO MAR2          03598000 D
0A77  4809 0F41 0020 00F0          BMAR L = BR1             % SAVE R(A)                        03599000 D
0A78  0000 0000 0000 0030          COMP 8 = SAR                                                 03600000 D
0A79  2F30 0000 0000 0060          EINPUT - 1 = CPCR        % (R(A)) INTO B                    03601000 D
0A7A  4809 0C41 2000 00F0          B L = A2                 % TEMP STORAGE OF (R(A))           03602000 D
0A7B  001C 0000 0000 00AC          COMP 16 = SAR; 1 = LIT                                       03603000 D
0A7C  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2       % R(A+1)                          03604000 D
0A7D  2F30 0000 0000 006C          EINPUT - 1 = CPCR        % (R(A+1)) INTO B                  03605000 D
0A7E  4809 CC5C 2030 00F0          A2 OR B = A2             % A2 = (R(A))/(R(A+1))             03606000 D
0A7F  4809 E156 0B00 00F0          A3 AND LIT = B           % ISOLATE "M" FIELD                03607000 D
0A80  00F0 0000 0000 00EC          15 = LIT                                                     03608000 D
0A81  51C0 0000 0000 0060          REGSTACK - 1 = CPCR      % R(M) INTO MAR2                   03609000 D
0A82  2F30 0000 0000 0060          EINPUT - 1 = CPCR        % (R(M)) INTO B                    03610000 D
0A83  4809 0C40 0030 80F0          B = SAR                  % (R(M)) (C-5) INTO SAR            03611000 D
0A84  49C9 C000 AB30 C0F0          A2 R = A2,B; SET LC1     % A2 HAS SHIFTED (R(A))/(R(A+1))   03612000 D
0A85  200C 0000 0000 C060          SETCCA - 1 = CPCR        % SET THE CONDITION BITS           03613000 D
0A86  4809 00C0 0000 20F0          ASR                      % REFERENCE BR1                    03614000 D
0A87  4809 0F43 8C1C 00F0          BMAR R = MAR2            % ADDR OF R(A) INTO MAR2           03615000 D
0A88  0000 00C0 0000 0010          B = SAR                                                      03616000 D
0A89  4809 C000 8030 00F0          A2 R = MIR               % (R(A)) INTO MIR                  03617000 D
0A8A  0000 0000 0000 0020          16 = SAR                                                     03618000 D
0A8B  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR       % WRITE OUT SHIFTED (R(A))         03619000 D
0A8C  4809 C001 2030 00F0          A2 L = A2                                                    03620000 D
0A8D  0010 0000 0000 00AC          COMP 16 = SAR; 1 = LIT                                       03621000 D
0A8E  4809 C000 8030 00F0          A2 R = MIR               % (R(A+1)) INTO MIR                03622000 D
0A8F  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2       % R(A+1)                          03623000 D
0A90  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR                                           03624000 D
0A91  66E0 00C0 0000 0040          OPCODE - 1 = MPCR                                            03625000 D
                                                            %                                    03626000 D
                                                            %                                    03627000 D
                                OP121:                      % RI TYPE 2 STORE DOUBLE              03628000 D
                                                            % (R(A),R(A+1)) INTO Y*, Y*+1        03629000 D
0A92  22B0 00C0 0000 0060          CONTENTSRA - 1 = CPCR    % (R(A)) INTO B, R(A) INTO MS WORD A3 03630000 D
0A93  4809 0C40 0030 00FC          B = MIR                  % TEMP STORAGE FOR (R(A))          03631000 D
0A94  238C 00C0 0000 C060          CONTENTSRM - 1 = CPCR    % (R(M)) INTO B                   03632000 D
0A95  4809 0C41 0C1C C0F0          B L = BR2                                                    03633000 D
0A96  0000 0000 0000 C030          COMP 8 = SAR                                                 03634000 D
0A97  61E0 00C0 0000 0060          EMULOUT - 1 = CPCR       % (R(A)) INTO Y*                  03635000 D
0A98  4809 0F46 0030 00FC          BMAR + 1 = MIR           % TEMP HOLDER FOR Y* + 1          03636000 D
0A99  4809 E000 9000 40F0          A3 R = A3, CSAR                                              03637000 D
0A9A  0CF0 0000 0000 00A0          16 = SAR; 15 = LIT                                           03638000 D
0A9B  4809 E00C 001C 00F0          A3 OR 1 = MAR2           % R(A+1)                          03639000 D
0A9C  2F3C 00C0 0000 C060          EINPUT - 1 = CPCR        % (R(A+1)) INTO B                  03640000 D
0A9D  4809 0C40 1000 00F0          B = A3                   % TEMP STORAGE OF (R(A))           03641000 D
0A9E  4809 00C0 0030 00F0          BMI                                                          03642000 D
0A9F  4809 E0C0 0C30 00F0          A3 = MIR                                                     03643000 D
0AA0  4809 0C41 0C1C C0F0          B L = BR2                % ADDR OF Y* + 1                   03644000 D
0AA1  0000 0000 0000 C030          COMP 8 = SAR                                                 03645000 D
0AA2  61EC 0000 0C30 C060          EMULOUT - 1 = CPCR                                           03646000 D
0AA3  66EC 00C0 000C C040          OPCODE - 1 = MPCR                                            03647000 D
                                                            %                                    03648000 D
                                                            %                                    03649000 D
                                OP122:                      % RK TYPE LOGICAL RIGHT DOUBLE SHIFT  03650000 D
                                                            % SHIFT (R(A), R(A+1)) RIGHT Y(0-5)  03651000 D
                                                            % ZEROFILL AND SET CC               03652000 D
0AA4  633C 0000 0000 0060          IFETCH - 1 = CPCR        % "Y" INTO B                       03653000 D
0AA5  4809 0C41 0B00 00F0          B L = B                                                      03654000 D
0AA6  0CF0 00C0 0000 C0A0          16 = SAR; 15 = LIT                                           03655000 D
0AA7  4809 EC5C 1000 00F0          A3 OR B = A3             % A3 HOLDS "Y"/INSTRUCTION         03656000 D
0AA8  4809 E156 0B00 00F0          A3 AND LIT = B           % ISOLATE "M" FIELD                03657000 D
```

```
0AA9  4809 0C52 0000 00FC        B EQL 0
0AAA  6809 00C0 0030 00F0        IF TRUE THEN SET LC1; SKIP ELSE STEP                    03659C00 D
0AAB  238C 00C0 0C30 0060        CONTENTSRM - 1 = CPCR   % (R(M)) INTO B                 03660C00 D
0AAC  4809 E000 A000 00F0        A3 R = A2               % "Y" INTO A2                    03661C00 D
0AAD  0000 0000 0000 0020        15 = SAR                                                 03662C00 D
0AAE  4809 CC40 2000 00F0        A2 + B = A2             % Y INTO A2                      03663C00 D
0AAF  4809 E000 8800 0CFC        A3 R = B                                                 03664C00 D
0AB0  8000 0000 0000 000C        4 = SAR                                                  03665C00 D
0AB1  4809 2C56 0030 00F0        LIT AND B = B           % ISOLATE "A" FIELD              03666C00 D
0AB2  00F0 00C0 0030 00E0        15 = LIT                                                 03667C00 D
0AB3  51CC 0003 0000 0060        REGSTACK - 1 = CPCR                                      03668C00 D
0AB4  2F3C 00C0 0030 C060        EINPUT - 1 = CPCR       % (R(A)) INTO B                 03669C00 D
0AB5  4809 0F41 002C 00F0        BMAR L = BR1            % SAVE ADDR OF R(A) IN BR1       03670C00 D
0AB6  0000 00C0 0030 C030        COMP B = SAR                                             03671C00 D
0AB7  4809 0C41 1000 00F0        6 L = A3                % (R(A)) INTO MS A3              03672C00 D
0AB8  0010 0003 0000 00A0        16 = SAR; 1 = LIT                                        03673C00 D
0AB9  4809 2F5C 001C 00F0        LIT OR BMAR = MAR2      % R(A+1)                         03674C00 D
0ABA  2F3C 0003 0000 0060        EINPUT - 1 = CPCR       % (R(A+1)) INTO B                03675C00 D
0ABB  4809 EC5C 0B30 00F0        A3 OR B = B             % D = (R(A))/(R(A+1))            03676C00 D
0ABC  4809 C000 0000 80F0        A2 = SAR                % Y (C-5) INTO SAR               03677C00 D
0ABD  49C9 CC40 AB00 00F0        B R = B, A2; SET LC1    % PERFORM RIGHT SHIFT OF Y (C-5) 03678C00 D
0ABE  2000 00C0 0030 C060        SETCCA - 1 = CPCR       % SET THE CONDITION BITS         03679C00 D
0ABF  4809 C000 8C30 00F0        A2 R = MIR              % SHIFTED (R(A)) INTO MIR        03680C00 D
0AC0  0000 00C0 0000 0020        16 = SAR                                                 03681C00 D
0AC1  4809 00C0 0000 20F0        ASR                     % REFERENCE BR1                  03682C00 D
0AC2  4809 0F43 801C 00F0        BMAR R = MAR2           % ADDR OF R(A)                   03683C00 D
0AC3  0000 0003 0C00 C010        8 = SAR                                                  03684C00 D
0AC4  2F50 00C0 0030 C060        EOUTPUT - 1 = CPCR      % WRITE NEW (R(A))               03685C00 D
0AC5  4809 2F5C 001C 00F0        LIT OR BMAR = MAR2      % R(A+1)                         03686C00 D
0AC6  001C 0003 0030 00A0        1 = LIT; 16 = SAR                                        03687C00 D
0AC7  4809 CD01 2000 00FC        A2 L = A2               % ISOLATE (R(A+1))               03688C00 D
0AC8  4809 CCC0 8C30 0CFC        A2 R = MIR                                               03689C00 D
0AC9  2F50 0000 0000 0060        EOUTPUT - 1 = CPCR        % WRITE NEW (R(A+1))           03690C00 D
0ACA  66E0 00C0 0C00 0040        OPCODE - 1 = MPCR                                        03691C00 D
                                                         %                                 03692C00 D
                                                         %                                 03693C00 D
                          OP123:                         % STORE DOUBLE                    03694C00 D
                                                         % STORE (R(A)) INTO Y, (R(A+1)) INTO Y+1 03695C00 D
0ACB  5700 0003 0030 0060        RXMFIELD - 1 = CPCR       % Y INTO B                     03696C00 D
0ACC  4809 0C40 0030 00F0        B = MIR                 % TRANSFER Y TO MIR              03697C00 D
0ACD  4809 E000 9000 00F0        A3 R = A3                                                03698C00 D
0ACE  8000 0000 0000 0000        4 = SAR                                                  03699C00 D
0ACF  4809 E156 0030 C0F0        A3 AND LIT = B          % ISOLATE "A" FIELD              03700C00 D
0AD0  00FC 0003 0030 00EC        15 = LIT                                                 03701C00 D
0AD1  51CC 0000 0000 0060        REGSTACK - 1 = CPCR     % ADDR OF R(A) INTO MAR2        03702C00 D
0AD2  2F30 00C0 0030 C060        EINPUT - 1 = CPCR       % (R(A)) INTO E                 03703C00 D
0AD3  4809 0C40 0030 C0F0        B = MIR,EMI             % (R(A)) INTO MIR, Y INTO E      03704C00 D
0AD4  4809 0F40 1000 C0F0        BMAR = A3               % SAVE ADDR OF R(A) IN A3        03705C00 D
0AD5  4809 0C41 0010 C0F0        B L = BR2                                                03706C00 D
0AD6  0000 0000 0030 0030        COMP B = SAR                                             03707C00 D
0AD7  61E0 0000 0000 0060        EMULOUT - 1 = CPCR      % WRITE (R(A)) INTO Y            03708C00 D
0AD8  4809 2F5C 200C 00FC        LIT OR BMAR = A2        % Y + 1 ADDRESS                  03709C00 D
0AD9  0010 C000 0000 00E0        1 = LIT                                                  03710C00 D
0ADA  4809 E0DC 001C C0F0        A3 OR 1 = MAR2          % ADDR OF R(A+1)                 03711C00 D
0ADB  2F30 0000 0000 C060        EINPUT - 1 = CPCR       % (R(A+1)) INTO B                03712C00 D
0ADC  4809 0C40 0030 C0FC        B = MIR                                                  03713C00 D
0ADD  4809 C0F1 001C 00F0        A2 L = BR2              % ADDR OF Y + 1                  03714C00 D
0ADE  0000 0000 0000 0030        COMP B = SAR                                             03715C00 D
0ADF  61E0 0000 0000 C060        EMULOUT - 1 = CPCR                                       03716C00 D
0AE0  66E0 00C0 0C00 0040        OPCODE - 1 = MPCR                                        03717C00 D
```

```
                                                        %                                   03718000 D
                                                        %                                   03719000 D
                                OPCODE13:               % SHIFT                             03720000 D
0AE1  5F90 0000 0000 C060               XFCODE - 1 = CPCR   % "F" INTO A2                   03721000 D
0AE2  4809 C640 0030 00F0               A2 + AMPCR = AMPCR                                  03722000 D
0AE3  14FC 0000 0000 C0C0               OP13F - 1 = AMPCR                                   03723000 D
0AE4  4809 0000 0030 00FC               STEP                                                03724000 D
0AE5  4824 0000 0030 00F0               EXEC                                                03725000 D
                                                        %                                   03726000 D
                                                        %                                   03727000 D
0AE6  1500 0000 0000 0040       OP13F:  OP130 - 1 = MPCR                                    03728000 D
0AE7  3000 0000 0000 0040               FAULT - 1 = MPCR                                    03729000 D
0AE8  1510 0000 0030 0040               OP132 - 1 = MPCR                                    03730000 D
0AE9  1520 0000 0030 0040               OP133 - 1 = MPCR                                    03731000 D
                                                        %                                   03732000 D
                                                        %                                   03733000 D
                                OP130:                  % RR TYPE ALG RIGHT DOUBLE SHIFT     03734000 D
                                                        % SHIFT (R(A),R(A+1)) RIGHT R(M) (0-5) 03735000 D
                                                        % SIGN FILL AND SET CC               03736000 D
0AEA  4809 0C40 8030 00F0               B R = B                                             03737000 D
0AEB  80FC 0000 0030 0080               4 = SAR; 15 = LIT                                   03738000 D
0AEC  4809 2C56 0B00 00F0               LIT AND B = B       % ISOLATE "A" FIELD             03739000 D
0AED  51CC 0000 0030 0060               REGSTACK - 1 = CPCR     % ADDR OF R(A) INTO MAR2    03740000 D
0AEE  2F30 0000 0030 0060               EINPUT - 1 = CPCR   % (R(A)) INTO B                 03741000 D
0AEF  4809 0C41 2030 00F0               B L = A2            % TEMP STORAGE OF (R(A))         03742000 D
0AF0  0010 0000 0000 00A0               COMP 16 = SAR; 1 = LIT                              03743000 D
0AF1  4809 0F41 0020 00F0               BMAR L = BR1        % TEMP STORAGE OF R(A)           03744000 D
0AF2  0000 0000 0000 0030               COMP 8 = SAR                                        03745000 D
0AF3  4809 2F5C 001C 00FC               LIT OR BMAR = MAR2  % R(A+1) INTO MAR2              03746000 D
0AF4  2F30 0000 0000 C060               EINPUT - 1 = CPCR   % (R(A+1)) INTO B               03747000 D
0AF5  4809 CC5C 2030 00F0               A2 OR B = A2        % A2 = (R(A))/(R(A+1))           03748000 D
0AF6  4809 C000 0030 00FC               A2                                                  03749000 D
0AF7  4A49 0000 0030 00F0               IF MST THEN SET LC2                                 03750000 D
0AF8  4809 E156 0B00 00F0               A3 AND LIT = B      % ISOLATE "M"                   03751000 D
0AF9  00F0 0000 0000 00E0               15 = LIT                                            03752000 D
0AFA  51CC 0000 0000 C060               REGSTACK - 1 = CPCR                                 03753000 D
0AFB  2F30 00C0 0000 0060               EINPUT - 1 = CPCR   % (R(M)) INTO B                 03754000 D
0AFC  4809 0C40 0030 80F0               B = SAR             % (R(M)) (0-5) INTO SAR         03755000 D
0AFD  49C9 C000 A800 00F0               A2 R = A2,B; SET LC1  % A2 HAS RIGHT SHIFTED (R(A))/(R(A+1)) 03756000 D
0AFE  2000 0000 0000 C060               SETCCA - 1 = CPCR   % SET THE CONDITION BITS        03757000 D
0AFF  3C19 0013 0B00 00F0               IF LC2 THEN B111 L = B; SKIP                        03758000 D
0B00  4809 0000 0B00 C0FC               0 = B                                               03759000 D
0B01  4809 CC5C 2030 20F0               A2 OR B = A2; ASR                                   03760000 D
0B02  4809 0F43 801C 00F0               BMAR R = MAR2       % ADDR OF R(A) INTO MAR2        03761000 D
0B03  0000 0000 0030 0010               8 = SAR                                             03762000 D
0B04  4809 C000 8030 C0F0               A2 R = MIR          % (R(A)) INTO MIR               03763000 D
0B05  0000 0000 0000 C020               16 = SAR                                            03764000 D
0B06  2F50 00C0 0030 0060               EOUTPUT - 1 = CPCR  % WRITE OUT SHIFTED (R(A))      03765000 D
0B07  4809 C001 2030 00F0               A2 L = A2                                           03766000 D
0B08  0010 0000 0000 00AC               COMP 16 = SAR; 1 = LIT                              03767000 D
0B09  4809 C000 8B30 00F0               A2 R = MIR, B       % SHIFTED (R(A+1)) INTO MIR      03768000 D
0B0A  4809 2F5C 001C 00F0               LIT OR BMAR = MAR2  % R(A+1)                        03769000 D
0B0B  2F50 0000 0000 0060               EOUTPUT - 1 = CPCR                                  03770000 D
0B0C  66E0 0000 0000 0040               OPCODE - 1 = MPCR                                   03771000 D
                                                        %                                   03772000 D
                                                        %                                   03773000 D
                                OP132:                  % PK TYPE ALG RIGHT DOUBLE SHIFT     03774000 D
                                                        % SHIFT (R(A),R(A+1)) RIGHT Y(0-5)   03775000 D
                                                        % SIGN FILL AND SET CC               03776000 D
0B0D  4809 2C56 0B00 00F0               LIT AND B = B                                       03777000 D
```

```
0F0E  00F0 0000 0000 C0EC        15 = LIT
0F0F  4809 0C52 0000 00F0        B EQL 0                  % CHECK "M" FOR 0          03779C00 D
0F10  6819 0000 0C00 00F0        IF TRUE THEN SKIP;                                  03780C00 D
0F11  2380 00C0 0C00 006C        CONTENTSRM - 1 = CPCR    % (R(M)) INTO B            03781C00 D
0F12  4809 0C41 0B00 00F0        B L = B                                             03782C00 D
0F13  0000 00C0 0C00 0020        16 = SAR                                            03783C00 D
0F14  4809 EC5C 1C0C 00F0        A3 OR B = A3             % A3 = (R(M))/INSTRUCTION   03784C00 D
0F15  6330 0000 0C00 006C        IFETCH - 1 = CPCR        % "Y" INTO B               03785C00 D
0F16  4809 E0C0 AC00 00F0        A3 R = A2                % (R(M)) INTO A2            03786C00 D
0F17  0000 00C0 0C00 0020        16 = SAR                                            03787C00 D
0F18  4809 CC40 2C00 C0F0        A2 + B = A2              % Y INTO A2                 03788C00 D
0F19  4809 E0C0 8B00 00F0        A3 R = B                 % ISOLATE "A"               03789C00 D
0F1A  90F0 00C0 0C00 0080        4 = SAR; 15 = LIT                                   03790C00 D
0F1B  4809 2C56 0B00 00F0        LIT AND B = B            % "A" INTO B               03791C00 D
0F1C  51C0 00C0 0C00 0060        REGSTACK - 1 = CPCR      % ADDR OF R(A) INTO MAR2   03792C00 D
0F1D  2F30 0000 0C00 006C        EINPUT - 1 = CPCR        % (R(A)) INTO B            03793C00 D
0F1E  4809 0C41 1C00 00F0        R L = A3                 % TEMP STORAGE OF (R(A))    03794C00 D
0F1F  0000 00C0 0C00 0020        COMP 16 = SAR                                       03795C00 D
0F20  4809 CF41 0C20 C0F0        BMAR L = BR1             % SAVE ADDR OF R(A) IN BR1  03796C00 D
0F21  001C 0000 0C00 0080        COMP 8 = SAR ; 1 = LIT                              03797C00 D
0F22  4809 2F5C 001C C0F0        LIT OR BMAR = MAR2       % R(A+1)                   03798C00 D
0F23  2F30 00C0 0C00 0060        EINPUT - 1 = CPCR        % (R(A+1)) INTO B          03799C00 D
0F24  4809 EC5C 1C00 00F0        A3 OR B = A3             % A3 = (R(A)/(R(A+1))       03800C00 D
0F25  4809 E0C0 0000 00F0        A3                                                  03801C00 D
0F26  4A49 0000 0000 00FC        IF MST THEN SET LC2                                 03802C00 D
0F27  4809 C0C3 0000 80F0        A2 = SAR                 % Y(0-5) INTO SAR          03803C00 D
0F28  49C9 E0C3 9B00 00F0        A3 R = A3,B; SET LC1     % SHIFT (R(A))/(R(A+1))     03804C00 D
0F29  2000 0000 0C00 0060        SETCCA - 1 = CPCR        % SET THE CONDITION BITS    03805C00 D
0F2A  3C19 0019 0B00 00F0        IF LC2 THEN B111 L = B; SKIP                        03806C00 D
0F2B  4809 0000 0B00 00F0        0 = B                                               03807C00 D
0F2C  4809 EC5C 1C00 00F0        A3 OR B = A3             % PERFORM SIGN FILL         03808C00 D
0F2D  4809 00C0 0000 20F0        ASR                      % REFERENCE BR1            03809C00 D
0F2E  4809 0F43 801C 00F0        BMAR R = MAR2            % ADDR OF R(A) INTO MAR2    03810C00 D
0F2F  0000 00C0 0C00 C01C        8 = SAR                                             03811C00 D
0F30  4809 E0C3 8030 00F0        A3 R = MIR                                          03812C00 D
0F31  0000 0000 0000 002C        16 = SAR                                            03813C00 D
0F32  2F50 00C0 0000 C060        EOUTPUT - 1 = CPCR       % WRITE OUT NEW (R(A))     03814C00 D
0F33  0010 0000 0C00 00A0        COMP 16 = SAR; 1 = LIT                              03815C00 D
0F34  4809 2F5C 001C 00FC        LIT OR BMAR = MAR2       % R(A+1)                   03816C00 D
0F35  4809 E001 1000 00F0        A3 L = A3                                           03817C00 D
0F36  4809 E000 8B90 00F0        A3 R = MIR, B            % NEW (R(A+1))              03818C00 D
0F37  2F50 0000 0C00 0060        EOUTPUT - 1 = CPCR                                  03819C00 D
0F38  66E0 0000 0000 C040        OPCODE - 1 = MPCR                                   03820C00 D
                                                          %                          03821C00 D
                                                          %                          03822C00 D
                                 OP133:                   % RX TYPE STORE MULTIPLE    03823C00 D
                                                          % (R(A),...R(M)) INTO Y,...Y+M-A+1  03824C00 D
0F39  6330 00C0 0000 0060        IFETCH - 1 = CPCR        % Y INTO B                 03825C00 D
0F3A  4809 0C41 C020 00FC        B L = BR1                % SAVE Y IN BR1             03826C00 D
0F3B  0000 00C0 0C00 003C        COMP 8 = SAR                                        03827C00 D
0F3C  4809 E156 2000 00FC        A3 AND LIT = A2          % "M" INTO A2               03828C00 D
0F3D  80F0 0000 0C00 0080        4 = SAR; 15 = LIT                                   03829C00 D
0F3E  4809 E000 9C00 C0F0        A3 R = A3                                           03830C00 D
0F3F  4809 E155 0B90 00F0        A3 AND LIT = B, MIR      % "A" INTO B, MIR          03831C00 D
0F40  4809 CC5E 0B0C 00F0        A2 - B = B               % CHECK IF "M" < "A"        03832C00 D
0F41  4419 0C46 0B00 C0F0        IF NOT MST THEN B + 1 = B; SKIP                     03833C00 D
0F42  4809 2C80 0B0C 00F0        LIT + B = B              % CALCULATE 16 + (M-A) + 1  03834C00 D
0F43  0110 00C0 0000 00E0        17 = LIT                                            03835C00 D
0F44  4809 0C5F 1000 C0F0        -B L = A3, BMI           % A3 IS NEG CTR, B HOLDS "A"  03836C00 D
0F45  000C 00C0 0000 0020        COMP 16 = SAR                                       03837C00 D
```

```
0846  51C0 0000 0000 0060   STMU:   REGSTACK - 1 = CPCR    % ADDR OF R(A) INTO MAR2                03838000 D
0847  2F3C 00C0 0000 C060           EINPUT - 1 = CPCR     % (R(A)) INTO B                         03839000 D
0848  4809 0F46 001C 00F0           BMAR + 1 = MAR2       % INCREMENT R(A)                         03840000 D
0849  4809 E0C0 9000 00F0           A3 R = A3             % CLEAR LHW OF A3                        03841000 D
084A  0000 00C0 0000 0020           16 = SAR                                                        03842000 D
084B  4809 E001 1C00 00F0           A3 L = A3                                                       03843000 D
084C  4809 EF5C 1000 00F0           A3 OR BMAR = A3       % A3 = -CTR/R(A+X)                       03844000 D
084D  4809 0C40 0030 20F0           B = MIR, ASR                                                    03845000 D
084E  4809 0F43 0010 00F0           BMAR = BR2            % LOAD BR2 WITH Y                        03846000 D
084F  61E0 0000 0030 0060           EMULOUT - 1 = CPCR    % WRITE OUT NEW Y VALUE                  03847000 D
0850  4809 0F46 001C 00F0           BMAR + 1 = MAR2       % NEXT Y ADDRESS                         03848000 D
0851  4809 0F4L 0020 00F0           BMAR L = BR1          % SAVE NEXT Y IN BR1                     03849000 D
0852  0000 00C0 0030 0030           COMP B = SAR                                                    03850000 D
0853  4809 00C1 0B30 00F0           1 L = B               % 1 IN BIT 16 OF B                       03851000 D
0854  0000 0000 0030 0020           COMP 16 = SAR                                                   03852000 D
0855  4809 EC40 1000 00F0           A3 + B = A3           % INCREMENT NEGATIVE COUNTER             03853000 D
0856  4809 E156 0B00 00F0           A3 AND LIT = B        % ISOLATE ADDR OF NEXT R(A+X)            03854000 D
0857  01F0 0000 0000 00E0           31 = LIT                                                        03855000 D
0858  4809 E0C0 0030 00F0           A3                    % A3 TO THE ADDER                        03856000 D
0859  480B 00C0 0030 00F0           IF MST THEN STEP ELSE SKIP  % IF CTR NEG, FETCH NEXT R(A+X)   03857000 D
085A  845C 0000 0030 004C           STMU - 1 = MPCR       % RETURN TO TOP OF LOOP                  03858000 D
085B  56E0 0000 0000 C040           OPCODE - 1 = MPCR                                               03859000 D
                                                          %                                       03860000 D
                                                          %                                       03861000 D
                                    OPCODE14:             % ALGEBRAIC LEFT SHIFT                   03862000 D
085C  5F9C 0000 0030 0060           XFCODE - 1 = CPCR     % "F" INTO A2                            03863000 D
085D  4809 C640 003C 00F0           A2 + AMPCR = AMPCR                                              03864000 D
085E  153C 00C0 0030 00C0           OP14F - 1 = AMPCR                                               03865000 D
085F  4809 00C0 0030 00F0           STEP                                                            03866000 D
0860  4824 0000 0000 00F0           EXEC                                                            03867000 D
                                                          %                                       03868000 D
                                                          %                                       03869000 D
0861  1540 0C00 0030 0040   OP14F:  OP140 - 1 = MPCR                                                03870000 D
0862  3000 00C0 0030 0040           FAULT - 1 = MPCR                                                03871000 D
0863  1550 00C0 0C00 0040           OP142 - 1 = MPCR                                                03872000 D
0864  156C 0000 0000 C04C           OP143 - 1 = MPCR                                                03873000 D
                                                          %                                       03874000 D
                                                          %                                       03875000 D
                                    OP140:                % RR TYPE ALG LEFT SHIFT (REGISTER)      03876000 D
                                                          % SHIFT (R(A))  LEFT (R(M)) (0-5) PLACES 03877000 D
                                                          % ZERO FILL, SET CC AND SET OVERFLOW BIT 03878000 D
0865  2380 0003 0000 0060           CONTENTSRM - 1 = CPCR % (R(M)) INTO B                         03879000 D
0866  4809 0C40 0030 C0F0           B = MIR               % TEMP STORAGE                           03880000 D
0867  4809 E0C0 0B0C 00F0           A3 = B                % PUT INSTR BACK INTO B                  03881000 D
0868  2280 00C3 0000 0060           CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                        03882000 D
0869  4809 0C40 2030 00F0           B = A2, BMI           % (R(A)) INTO A2, (R(M)) INTO B          03883000 D
086A  4809 2C55 0B30 00F0           LIT AND B = B         % MASK BITS 0-5                          03884000 D
086B  03FC 0000 0030 C0E0           63 = LIT                                                        03885000 D
086C  4809 2C59 0030 00FC           LIT LEQ B             % CHECK FOR SHIFT >= 16                 03886000 D
086D  010C 0003 0000 00E0           16 = LIT                                                        03887000 D
086E  7819 0000 0000 00F0           IF FALSE THEN SKIP                                              03888000 D
086F  157C 00C0 0030 0040           C140 - 1 = MPCR       % SHIFT >= 16                           03889000 D
0870  4809 0C5E 0030 80FC           -B = SAR                                                        03890000 D
0871  4809 C001 2000 C0F0           A2 L = A2             % PERFORM LEFT SHIFT                     03891000 D
0872  4809 C0C3 9000 00FC           A2 R = A3             % PUT DIGITS SHIFTED OUT IN A3           03892000 D
0873  000C 0000 0030 0020           16 = SAR                                                        03893000 D
0874  4809 C001 2030 C0FC           A2 L = A2             % CLEAR UHW A2                           03894000 D
0875  4809 C0C0 8B30 C0FC           A2 R = MIR, B         % SHIFTED (R(A)) INTO MIR                03895000 D
0876  4809 E0C2 0030 00FC           NOT A3                                                          03896000 D
0877  63C9 0000 0000 C0F0           IF NOT ABT THEN SET LC1  % FLAG FOR SETTING OV BIT              03897000 D
```

```
0B78  5020 0000 0030 0060          OVBIT - 1 = CPCR        % SET OV BIT ACCORDINGLY          03898000 D
0B79  2000 0000 0000 0060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS           03899000 D
0B7A  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                        03900000 D
0B7B  66E0 0000 0030 0040          OPCODE - 1 = MPCR                                         03901000 D
0B7C  4809 C012 0030 00F0   C140:  A2 EQL 0                % CHECK FOR 0 IN R(A)              03902000 D
0B7D  63C9 0000 0000 00F0          IF FALSE THEN SET LC1                                     03903000 D
0B7E  5020 0000 0000 0060          OVBIT - 1 = CPCR        % SET OV BIT ACCORDINGLY          03904000 D
0B7F  4809 0000 0030 00F0          0 = MIR                 % SHIFT >= 16, -> 0  INTO R(A)     03905000 D
0B80  21B0 0000 0030 0060          SETOG - 1 = CPCR                                          03906000 D
0B81  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR                                        03907000 D
0B82  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                         03908000 D
                                                           %                                 03909000 D
                                                           %                                 03910000 D
                            OP142:                         % RR TYPE ALG LEFT SINGLE SHIFT    03911000 D
                                                           % SHIFT (R(A)) LEFT Y (0-5) PLACES 03912000 D
                                                           % ZERO FILL AND SET CC, SET OVERFLOW BIT 03913000 D
0B83  6330 0000 0030 0060          IFETCH - 1 = CPCR       % "Y" INTO B                       03914000 D
0B84  4809 0040 0030 00F0          B = MIR                 % "Y" INTO MIR                     03915000 D
0B85  4809 E156 0B30 00F0          A3 AND LIT = B          % ISOLATE "M"                      03916000 D
0B86  00F0 0000 0000 00E0          15 = LIT                                                  03917000 D
0B87  4809 0052 0000 00F0          0 EQL B                 % CHECK FOR 0 , "M" FIELD         03918000 D
0B88  6819 0000 0030 00F0          IF TRUE THEN SKIP                                         03919000 D
0B89  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                    03920000 D
0B8A  4809 0040 2030 00F0          B = A2, BMI             % (R(M) INTO A2, "Y" INTO B        03921000 D
0B8B  4809 C040 0030 00F0          A2 + B = MIR            % Y INTO MIR                       03922000 D
0B8C  4809 E000 9000 00F0          A3 R = A3                                                 03923000 D
0B8D  80F0 0000 0030 008C          4 = SARI 15 = LIT                                         03924000 D
0B8E  4809 E156 0B00 00F0          A3 AND LIT = B          % ISOLATE "A"                     03925000 D
0B8F  51C0 0000 0000 0060          REGSTACK - 1 = CPCR     % R(A) INTO MAR2                  03926000 D
0B90  2F30 0000 0000 0060          EINPUT - 1 = CPCR       % (R(A)) INTO B                   03927000 D
0B91  4809 0040 2030 00F0          B = A2, BMI             % (R(A)) INTO A2, Y INTO B         03928000 D
0B92  4809 2056 0B00 00F0          LIT AND B = B           % ISOLATE SHIFT AMOUNT             03929000 D
0B93  03F0 0000 0000 00E0          63 = LIT                                                  03930000 D
0B94  4809 2059 0030 00F0          LIT LEQ B               % IS SHIFT AMOUNT > 16 ?           03931000 D
0B95  0100 0000 0030 00E0          16 = LIT                                                  03932000 D
0B96  7819 0000 0000 00F0          IF FALSE THEN SKIP                                        03933000 D
0B97  158C 0000 0000 0040          C142 - 1 = MPCR         % SHIFT > 16                      03934000 D
0B98  4809 0055 0000 80F0          -8 = SAR                                                  03935000 D
0B99  4809 C001 2000 00F0          A2 L = A2               % PERFORM REQUIRED LEFT SHIFT      03936000 D
0B9A  4809 C000 9000 00F0          A2 R = A3               % PUT LEFT SHIFTED BITS INTO A3    03937000 D
0B9B  0000 0000 0000 0020          16 = SAR                                                  03938000 D
0B9C  4809 C001 2030 00F0          A2 L = A2               % CLEAR UHW OF A2                  03939000 D
0B9D  4809 C000 8B30 00F0          A2 R = MIR, B           % NEW (R(A)) INTO MIR, B           03940000 D
0B9E  4809 E002 0000 00F0          NOT A3                                                    03941000 D
0B9F  63C9 0000 0000 00F0          IF NOT ABT THEN SET LC1                                   03942000 D
0BA0  5020 0000 0000 0060          OVBIT - 1 = CPCR        % SET THE OV BIT ACCORDINGLY      03943000 D
0BA1  2000 0000 0000 0060          SETCCA - 1 = CPCR       % SET THE CC                      03944000 D
0BA2  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR      % WRITE OUT NEW R(A)              03945000 D
0BA3  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                         03946000 D
0BA4  4809 C012 0030 00F0   C142:  A2 EQL 0                % CHECK FOR A ZERO (R(A))          03947000 D
0BA5  63C9 0000 0000 00F0          IF FALSE THEN SET LC1                                     03948000 D
0BA6  5020 0000 0030 0060          OVBIT - 1 = CPCR        % SET THE OV BIT                  03949000 D
0BA7  4809 0000 0030 00F0          0 = MIR                 % PREPARE TO WRITE 0 INTO R(A)     03950000 D
0BA8  21B0 0000 0030 0060          SETOG - 1 = CPCR                                          03951000 D
0BA9  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR                                        03952000 D
0BAA  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                         03953000 D
                                                           %                                 03954000 D
                                                           %                                 03955000 D
                            OP143:                         % RX TYPE BYTE STORE INDEX         03956000 D
                                                           % (R(A)) BITS 0-7 INTO Y BYTE      03957000 D
```

```
0BAB  4849 0003 0030 00F0                SET LC2                    % BYTE INSTRUCTION FLAG                03959000 D
0BAC  5700 0000 0000 0060                RXMFIELD - 1 = CPCR        % Y ADDR INTO E                        03960000 D
0BAD  4809 0C41 0030 00F0                B L = BR1                  % TEMP STORAGE OF Y ADDR               03961000 D
0BAE  0000 0003 0000 0030                COMP 8 = SAR                                                     03962000 D
0BAF  4809 E000 0B00 00F0                A3 = B                                                           03963000 D
0BB0  22B0 0000 0000 0060                CONTENTSRA - 1 = CPCR      % (R(A)) INTO B                       03964000 D
0BB1  4809 2C55 0030 00F0                LIT AND B = MIR            % (R(A)) BITS 0-7 INTO MIR             03965000 D
0BB2  0FF0 0000 0030 00E0                255 = LIT                                                        03966000 D
0BB3  2380 0000 0000 0060                CONTENTSRM - 1 = CPCR      % (R(M)) INTO B                       03967000 D
0BB4  4809 0C45 0B30 00F0                B + 1 = B                                                        03968000 D
0BB5  4809 0C40 0030 00F0                B = MIR, BMI               % (R(M)) + 1 INTO MIR, (R(A)) INTO B   03969000 D
0BB6  2F50 0000 0000 0060                EOUTPUT - 1 = CPCR         % (R(M)) + 1 INTO R(M)                03970000 D
0BB7  4809 0C40 1030 00F0                B = A3                     % MOVE (R(A)) BITS 0-7 TO LS BYTE OF A3 03971000 D
0BB8  4809 0000 0000 20F0                ASR                        % REFERENCE BR1                        03972000 D
0BB9  4809 0F40 0010 00F0                BMAR = BR2                 % Y ADDR INTO ER2                      03973000 D
0BBA  60E0 0000 0030 0060                EMULIN - 1 = CPCR          % (Y) INTO B                           03974000 D
0BBB  380B 0000 0000 00F0                IF LC2 THEN STEP ELSE SKIP % IF LC2, PUT BYTE INTO LS BYTE       03975000 D
0BBC  159C 0000 0000 0060                LS143 - 1 = CPCR                                                 03976000 D
0BBD  4809 0C41 0B30 40F0                B L = B, CSAR                                                    03977000 D
0BBE  0000 0000 0030 0010                COMP 24 = SAR                                                    03978000 D
0BBF  4809 0C40 8B00 00F0                B R = B                                                          03979000 D
0BC0  4809 E001 1030 00F0                A3 L = A3                                                        03980000 D
0BC1  0000 0000 0000 0030                COMP 8 = SAR                                                     03981000 D
0BC2  4809 EC5C 0030 00F0                A3 OR B = MIR                                                    03982000 D
0BC3  4809 0F41 0010 00F0                BMAR L = BR2               % SET UP BR2 FOR EMULOUT               03983000 D
0BC4  0000 0000 0000 0030                COMP 8 = SAR                                                     03984000 D
0BC5  61E0 0000 0000 0060                EMULOUT - 1 = CPCR                                               03985000 D
0BC6  66E0 0000 0000 0040                OPCODE - 1 = MPCR                                                03986000 D
0BC7  4809 0C40 8B30 40F0        LS143:  B R = B, CSAR              % CLEAR LS BYTE OF B                   03987000 D
0BC8  4809 0C41 0B00 00F0                B L = B                                                          03988000 D
0BC9  4809 EC5C 0030 00F0                A3 OR B = MIR                                                    03989000 D
0BCA  4809 0F41 0010 00F0                BMAR L = BR2               % SET UP FOR EMULOUT                   03990000 D
0BCB  61E0 0000 0000 0060                EMULOUT - 1 = CPCR                                               03991000 D
0BCC  66E0 0000 0000 0040                OPCODE - 1 = MPCR                                                03992000 D
                                                                    %                                     03993000 D
                                                                    %                                     03994000 D
0BCD  5F90 0000 0000 0060        OPCODE15:  XFCODE - 1 = CPCR       % "F" INTO A2                          03995000 D
0BCE  4809 C640 0030 00F0                A2 + AMPCR = AMPCR                                               03996000 D
0BCF  15A0 0000 0000 0000                OP15F - 1 = AMPCR                                                03997000 D
0BD0  4809 0000 0000 00F0                STEP                                                             03998000 D
0BD1  4824 0000 0000 00F0                EXEC                                                             03999000 D
                                                                    %                                     04000000 D
0BD2  15BC 0000 0000 0040        OP15F:  OP150 - 1 = MPCR                                                 04001000 D
0BD3  15C0 0000 0000 0040                OP151 - 1 = MPCR                                                 04002000 D
0BD4  15D0 0000 0000 0040                OP152 - 1 = MPCR                                                 04003000 D
0BD5  15E0 0000 0000 0040                OP153 - 1 = MPCR                                                 04004000 D
                                                                    %                                     04005000 D
                                                                    %                                     04006000 D
                                 OP150:                             % RR TYPE CIRCULAR SINGLE SHIFT        04007000 D
                                                                    % SHIFT (R(A)) LEFT CIRCULARLY (R(M)) 04008000 D
                                                                    % BITS 0-5 AND SET CC                  04009000 D
0BD6  2380 0000 0000 0060                CONTENTSRM - 1 = CPCR      % (R(M)) INTO B                        04010000 D
0BD7  4809 0C40 0030 00F0                B = MIR                    % (R(M)) INTO MIR                      04011000 D
0BD8  4809 E000 0B00 00F0                A3 = B                                                           04012000 D
0BD9  22B0 0000 0000 0060                CONTENTSRA - 1 = CPCR      % (R(A)) INTO B                        04013000 D
0BDA  4809 0C40 2000 00F0                E = A2                                                           04014000 D
0BDB  4809 C001 2030 00F0                A2 L = A2                                                        04015000 D
0BDC  03F0 0000 0000 00A0                COMP 16 = SAR; 63 = LIT                                          04016000 D
0BDD  4809 CD5C 2030 00F0                A2 OR B = A2, BMI          % A2 = (R(A))/(R(A)), (R(M)) INTO E    04017000 D
```

```
0BDF  4812 0000 0000 00F0          SAVE                                                    04019C00 D
0BE0  4809 E15E 0000 00F0          A3 GEQ LIT                                              04020C00 D
0BE1  0100 0000 0030 00E0          16 = LIT                                                04021C00 D
0BE2  7C29 E15E 1000 00F0          IF TRUE THEN A3+ NOT LIT + 1 = A3; JUMP                 04022C00 D
0BE3  4809 E0C0 0000 00F0          A3 = B                                                  04023C00 D
0BE4  4809 2C5E 003C 80F0          LIT - B = SAR                                           04024C0C D
0BE5  4809 C001 8800 00F0          A2 C = B                                                04025C00 D
0BE6  4809 0C41 0B0C 00F0          B L = B                                                 04026C0C D
0BE7  000C 0000 0000 C020          16 = SAR                                                04027000 D
0BE8  4809 0C40 8B3C 00F0          B R = MIR, B                                            04028C00 D
0BE9  2F50 00C0 0030 0060          EOUTPUT - 1 = CPCR                                      04029C00 D
0BEA  2000 0000 C0CC C060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS         04030C00 D
0BEB  66E0 00C0 0000 0040          OPCODE - 1 = MPCR                                       04031C00 D
                                                          %                                04032C00 D
                                                          %                                04033C00 D
                                OP151:                    % RI TYPE 2 STORE AND INDEX BY 1  04034C00 D
                                                          % (R(A)) INTO Y*, (R(M)) + 1 INTO R(M)  04035C00 D
0BEC  2380 0C00 0030 006C          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                  04036C00 D
0BED  4809 0C41 0020 00F0          B L = BR1              % Y* INTO BR1                    04037C00 D
0BEE  000C 00C0 0C00 0030          COMP B = SAR                                            04038C00 D
0BEF  4809 E0C0 0B30 00F0          A3 = B                                                  04039C00 D
0BF0  2280 0C00 0000 C060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                  04040C0C D
0BF1  4809 0C40 0C30 20F0          B = MIR; ASR                                            04041C00 D
0BF2  4809 0F43 0010 00F0          BMAR = BR2                                              04042C00 D
0BF3  61E0 0000 0C00 006C          EMULOUT - 1 = CPCR     % (R(A)) INTO Y*                 04043C0C D
0BF4  4809 0F46 0C30 00F0          BMAR + 1 = MIR         % (R(M)) + 1                     04044C00 D
0BF5  4809 E156 0B30 00F0          A3 AND LIT = B         % ISOLATE "M" FIELD              04045C00 D
0BF6  00F0 00C0 0000 00E0          15 = LIT                                                04046C00 D
0BF7  51C0 00C0 0030 0060          REGSTACK - 1 = CPCR    % MAR2 = ADDR OF R(M)            04047C00 D
0BF8  2F50 00C0 0030 0060          EOUTPUT - 1 = CPCR     % (R(A)) + 1 INTO R(M)           04048C00 D
0BF9  66E0 00C0 0030 0040          OPCODE - 1 = MPCR                                       04049C00 D
                                                          %                                04050C00 D
                                                          %                                04051C00 D
                                OP152:                    % RK TYPE CIRCULAR LEFT SINGLE SHIFT  04052000 D
                                                          % SHIFT (R(A)) Y (C-5) PLACES AND SET CC  04053C00 D
0BFA  4809 2C56 0B3C 00F0          LIT AND B = B          % ISOLATE "M"                    04054C0C D
0BFB  00F0 00C0 0030 00E0          15 = LIT                                                04055C00 D
0BFC  4809 0C52 003C 00F0          B EQL 0                                                 04056C00 D
0BFD  6819 0000 0C00 C0F0          IF TRUE THEN SKIP                                       04057000 D
0BFE  2380 0000 0030 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                  04058C00 D
0BFF  4809 0C41 0B30 00FC          B L = B                                                 04059C00 D
0C00  000C 0000 0C00 0020          COMP 16 = SAR                                           04060C00 D
0C01  4809 EC5C 1000 00F0          A3 OR B = A3           % A3 = (R(M))/INSTRUCTION        04061C00 D
0C02  633C 00C0 0070 0060          IFETCH - 1 = CPCR      % Y INTO B                       04062C00 D
0C03  4809 EC00 AC0C 0CF0          A3 R = A2              % (R(M)) INTO A2                 04063C00 D
0C04  000C 0000 C03C 0020          16 = SAR                                                04064C00 D
0C05  4809 CC40 2000 00FC          A2 + B = A2            % Y INTO A2                      04065C0C D
0C06  4809 C156 0030 00F6          A2 AND LIT = MIR       % Y(C-5)                         04066C00 D
0C07  93F0 C000 0C0C 008C          63 = LIT; 4 = SAR                                       04067C00 D
0C08  4809 E0C3 900C C0F0          A3 R = A3                                               04068C00 D
0C09  4809 E156 0B30 00F0          A3 AND LIT = B         % ISOLATE "A"                   04069C00 D
0C0A  00FC 00C0 0000 C0EC          15 = LIT                                                04070000 D
0C0B  51C0 00C0 0C00 006C          REGSTACK - 1 = CPCR    % ADDR OF R(A) INTO MAR2         04071C00 D
0C0C  2F30 000D 0C00 006C          EINPUT - 1 = CPCR      % (R(A)) INTO B                  04072000 D
0C0D  4809 0C41 100C 00F0          B L = A3                                                04073C0C D
0C0E  0100 E0C0 0C00 00A0          COMP 16 = SAR; 16 = LIT                                 04074C00 D
0C0F  4809 EC5C 2000 00F0          A3 OR B = A2           % A2 = (R(A))/(R(A))             04075C0C D
0C10  4809 0000 0000 00F0          PMI                                                     04076C00 D
0C11  4809 0C40 1000 00FC          B = A3                                                  04077C00 D
```

```
0C12  4812 0000 0000 00F0                SAVE
0C13  4809 E15E 0000 00F0                A3 GEQ LIT            % CHECK FOR SHIFT > 16                    04079000 D
0C14  7C29 E15E 1000 00F0                IF TRUE THEN A3 + NOT LIT + 1 = A3; JUMP                        04080000 D
0C15  4809 E000 0B30 00FC                A3 = B                                                          04081000 D
0C16  4809 2C5E 0030 80F0                LIT - B = SAR                                                   04082000 D
0C17  4809 C0C1 8B30 00F0                A2 C = B                                                        04083000 D
0C18  4809 0C41 0B30 00F0                B L = B                                                         04084000 D
0C19  0000 0000 0000 0020                16 = SAR                                                        04085000 D
0C1A  4809 0C40 8B30 00F0                B R = MIR, B                                                    04086000 D
0C1B  2F50 0000 0000 0060                EOUTPUT - 1 = CPCR                                              04087000 D
0C1C  2000 0000 0000 0060                SETCCA - 1 = CPCR     % SET THE CONDITION BITS                  04088000 D
0C1D  56E0 0000 0000 0040                OPCODE - 1 = MPCR                                               04089000 D
                                                               %                                         04090000 D
                                                               %                                         04091000 D
                                OP153:                         % RX TYPE STORE AND INDEX BY 1             04092000 D
                                                               % (R(A)) INTO Y, (R(M)) + 1 -> R(M)        04093000 D
0C1E  5700 0000 0000 0060                RXMFIELD - 1 = CPCR   % ADDR OF Y INTO B                        04094000 D
0C1F  4809 0C41 0020 00F0                B L = BR1                                                       04095000 D
0C20  0000 0000 0030 003C                COMP 8 = SAR                                                    04096000 D
0C21  4809 E000 0B00 00F0                A3 = B                                                          04097000 D
0C22  2280 0000 0000 0060                CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                          04098000 D
0C23  4809 0C40 003C 20F0                B = MIR; ASR                                                    04099000 D
0C24  4809 0F43 0C1C 00F0                BMAR = BR2            % Y ADDR INTO ER2                         04100000 D
0C25  61E0 0000 0000 0060                EMULOUT - 1 = CPCR                                              04101000 D
0C26  2380 0000 0030 C060                CONTENTSFM - 1 = CPCR  % (R(M)) INTO B                          04102000 D
0C27  4809 0C46 0030 00F0                B + 1 = MIR                                                     04103000 D
0C28  2F50 0000 0030 0060                EOUTPUT - 1 = CPCR                                              04104000 D
0C29  66E0 0000 0000 0040                OPCODE - 1 = MPCR                                               04105000 D
                                                               %                                         04106000 D
                                                               %                                         04107000 D
                                OPCODE16:                                                                04108000 D
0C2A  5F90 0000 0030 C060                XFCODE - 1 = CPCR     % "F" INTO A2                             04109000 D
0C2B  4809 C640 0C40 C0F0                A2 + AMPCR = AMPCR                                              04110000 D
0C2C  15F0 0000 0030 00C0                OP16F - 1 = AMPCR                                               04111000 D
0C2D  4809 0000 0000 C0F0                STEP                                                            04112000 D
0C2E  4824 0000 0000 00F0                EXEC                                                            04113000 D
                                                               %                                         04114000 D
0C2F  160C 0000 0030 0040       OP16F:   OP160 - 1 = MPCR                                                04115000 D
0C30  1610 0000 0030 0040                OP161 - 1 = MPCR                                                04116000 D
0C31  1620 0000 0000 0040                OP162 - 1 = MPCR                                                04117000 D
0C32  163C 0000 0030 C040                OP163 - 1 = MPCR                                                04118000 D
                                                               %                                         04119000 D
                                OP160:                         % RR TYPE ALGEBRAIC LEFT DOUBLE SHIFT      04120000 D
                                                               % SHIFT (R(A),R(A+1)) LEFT (R(M)) (0-5)    04121000 D
                                                               % ZERO FILL AND SET CC, SET OVERFLOW      04122000 D
0C33  4809 0C40 9B00 00F0                B R = B                                                         04123000 D
0C34  80FC 0000 0000 0080                A = SAR; 15 = LIT                                               04124000 D
0C35  4809 2C56 0B30 00F0                LIT AND B = B         % ISOLATE THE "A" FIELD                   04125000 D
0C36  51C0 0000 0030 0060                REGSTACK - 1 = CPCR   % ADDR OF R(A) INTO MAR2                  04126000 D
0C37  2F30 0000 0000 006C                EINPUT - 1 = CPCR     % (R(A)) INTO B                          04127000 D
0C38  4809 0C41 2030 00F0                B L = A2              % TEMP STORAGE OF (R(A))                  04128000 D
0C39  001C 00C0 0030 C0AC                COMP 16 = SAR; 1 = LIT                                          04129000 D
0C3A  4809 0F41 0020 00F0                BMAR L = BR1          % TEMP STORAGE OF R(A)                    04130000 D
0C3B  0000 0000 0000 003C                COMP 8 = SAR                                                    04131000 D
0C3C  4809 2F5C 001C 00F0                LIT OR BMAR = MAR2    % R(A+1)                                 04132000 D
0C3D  2F3C 0000 0000 006C                EINPUT - 1 = CPCR     % (R(A+1)) INTO B                         04133000 D
0C3E  4809 C05C 2030 C0F0                A2 OR B = A2          % A2 = (R(A))/(R(A+1))                    04134000 D
0C3F  4809 E156 0B30 00F0                A3 AND LIT = B        % ISOLATE "H" FIELD                       04135000 D
0C40  00F0 0000 0030 00E0                15 = LIT                                                        04136000 D
0C41  51C0 0000 0000 0060                REGSTACK - 1 = CPCR                                             04137000 D
```

```
0C43  4809 0C5E 0030 80F0          -B = SAR                                    04139000 D
0C44  4809 C000 9070 C0F0          A2 R = A3              % FOR OVERFLOW TEST   04140000 D
0C45  4809 E002 0000 00F0          NOT A3                                      04141000 D
0C46  63C9 0000 0030 00F0          IF NOT ABT THEN SET LC1                     04142000 D
0C47  302C C000 0000 0060          OVBIT - 1 = CPCR       % SET OVERFLOW BIT   04143000 D
0C48  4809 0C5E 0030 80F0          -B = SAR                                    04144000 D
0C49  49C9 C001 2B0C 00F0          A2 L = A2, B; SET LC1                       04145000 D
0C4A  20DC 00C0 0030 006C          SETCCA - 1 = CPCR                           04146000 D
0C4B  4809 C0C0 0030 20F0          ASR                    % REFERENCE BR1       04147000 D
0C4C  4809 0F43 801C 00F0          BMAR R = MAR2                               04148000 D
0C4D  000C 00C0 0000 0010          B = SAR                                     04149000 D
0C4E  4809 C0C0 8030 00F0          A2 R = MIR                                  04150000 D
0C4F  00C0 0000 0C00 C020          16 = SAR                                    04151000 D
0C50  2F50 00C0 0C0C 0060          EOUTPUT - 1 = CPCR     % WRITE OUT SHIFTED (R(A)) 04152000 D
0C51  0010 00C0 0000 00A0          COMP 16 = SAR, 1 = LIT                      04153000 D
0C52  4809 C001 200C 00F0          A2 L = A2                                   04154000 D
0C53  4809 C0C0 8030 00F0          A2 R = MIR                                  04155000 D
0C54  48C9 2F5C 001C 00F0          LIT OR BMAR = MAR2     % R(A+1)             04156000 D
0C55  2F50 00C0 0030 0060          EOUTPUT - 1 = CPCR                          04157000 D
0C56  56EC 0000 0000 C040          OPCODE - 1 = MPCR                           04158000 D
                                                          %                    04159000 D
                                                          %                    04160000 D
                                   OP161:                 % RI TYPE 2 STORE AND INDEX BY 2 04161000 D
                                                          % (R(A),R(A+1)) INTO Y*, Y*+1 04162000 D
                                                          % (R(M)) + 2 INTO R(M)  04163000 D
0C57  238C 0000 C030 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B      04164000 D
0C58  4809 0C41 002C 00F0          B L = BR1              % Y* INTO CR1         04165000 D
0C59  002C 0000 0000 C0B0          COMP B = SAR, 2 = LIT                       04166000 D
0C5A  4809 2C40 0C9C 00F0          LIT + B = MIR                               04167000 D
0C5B  4809 0F41 0C1C 00F0          BMAR L = BR2                                04168000 D
0C5C  51E0 0000 0000 C060          EMULOUT - 1 = CPCR     % (R(M)) + 2 INTO R(M) 04169000 D
0C5D  48C9 E000 8800 00F0          A3 R = B                                    04170000 D
0C5E  80FC 0000 0000 0080          15 = LIT, 4 = SAR                           04171000 D
0C5F  4809 2C56 0B30 00F0          LIT AND B = B          % ISOLATE "A"        04172000 D
0C60  51C0 0000 0C00 0060          REGSTACK - 1 = CPCR    \                    04173000 D
0C61  4809 E0C0 6B30 00F0          A3 = B                                      04174000 D
0C62  22B0 0000 000C C060          CONTENTSRA - 1 = CPCR % (R(A)) INTO E       04175000 D
0C63  4809 0C40 009C 00F0          B = MIR                                     04176000 D
0C64  4809 00C0 0000 20F0          ASR                    % REFERENCE BR1       04177000 D
0C65  4809 0F43 0010 00F0          BMAR = BR2                                  04178000 D
0C66  61EC 00C0 0000 006C          EMULOUT - 1 = CPCR     % (R(A)) INTO Y*      04179000 D
0C67  48C9 0F47 0020 00F0          BMAR + 1 L = BR1       % ADDR OF Y* + 1      04180000 D
0C68  000C 00C0 0000 C030          COMP B = SAR                                04181000 D
0C69  4809 E0C0 801C 00F0          A3 R = MAR2            % ADDR OF R(A)        04182000 D
0C6A  001C 0000 0000 C0AC          16 = SAR, 1 = LIT                           04183000 D
0C6B  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2     % R(A+1)             04184000 D
0C6C  2F3C 00C0 0030 C060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B    04185000 D
0C6D  4809 0C40 009C 00F0          B = MIR                                     04186000 D
0C6E  4809 00C0 0C30 20F0          ASR                    % REFERENCE BR1       04187000 D
0C6F  4809 0F43 0010 00F0          BMAR = BR2                                  04188000 D
0C70  61EC 0000 0000 C060          EMULOUT - 1 = CPCR     % (R(A+1)) INTO Y* +1 04189000 D
0C71  66E0 0000 C030 0040          OPCODE - 1 = MPCR                           04190000 D
                                                          %                    04191000 D
                                                          %                    04192000 D
                                   OP162:                 % RK TYPE ALG LEFT DOUBLE SHIFT 04193000 D
                                                          % SHIFT (R(A),R(A+1) LEFT Y (0-5) PLACES 04194000 D
                                                          % ZERO FILL, SET CC AND SET OVERFLOW BIT 04195000 D
0C72  4809 2C56 0B30 00F0          LIT AND B = B          % ISOLATE "M"         04196000 D
0C73  00FC 0000 0000 00EC          15 = LIT                                    04197000 D
```

```
0C74  4809 0C52 0000 00F0          0 EQL B                                          04198000 D
0C75  6819 0000 0030 00F0          IF TRUE THEN SKIP                                04199000 D
0C76  2380 0000 0C00 0060          CONTENTSRM - 1 = CPCR   % (R(M)) INTO B          04200000 D
0C77  4809 0C41 0B30 00F0          B L = B                                          04201000 D
0C78  0000 0000 0000 0020          16 = SAR                                         04202000 D
0C79  4809 EC5C 1030 00F0          A3 OR B = A3            % A3 = (R(M))/INSTRUCTION 04203000 D
0C7A  6330 0000 0000 0060          IFETCH - 1 = CPCR       % "Y" INTO B             04204000 D
0C7B  4809 E000 AC30 00F0          A3 R = A2                                        04205000 D
0C7C  0000 0000 0030 0020          16 = SAR                                         04206000 D
0C7D  4809 CC40 C080 00F0          A2 + B = MIR            % Y INTO MIR              04207000 D
0C7E  4809 E000 9030 00F0          A3 R = A3                                        04208000 D
0C7F  50F0 0000 0000 0080          4 = SAR; 15 = LIT                                04209000 D
0C80  4809 E155 0B70 00F0          A3 AND LIT = B          % ISOLATE "A"            04210000 D
0C81  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                              04211000 D
0C82  4809 0F41 0020 C0F0          BMAR L = BR1            % TEMP STORAGE OF R(A)   04212000 D
0C83  0000 0000 0000 0030          COMP B = SAR                                     04213000 D
0C84  2F30 0000 0030 C060          EINPUT - 1 = CPCR       % (R(A)) INTO B          04214000 D
0C85  4809 0C41 1030 00F0          B L = A3                                         04215000 D
0C86  0010 0000 0030 00A0          COMP 16 = SAR; 1 = LIT                           04216000 D
0C87  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2      % R(A+1)                 04217000 D
0C88  2F30 0000 0030 0060          EINPUT - 1 = CPCR                                04218000 D
0C89  4809 EC5C 2000 00F0          A3 OR B = A2, BMI       % A2 = (R(A))/(R(A+1)), B = Y 04219000 D
0C8A  4809 0C5E 0030 80F0          -B = SAR                % SET UP SHIFT AMOUNT     04220000 D
0C8B  4809 C000 9000 00F0          A2 R = A3               % BITS LOST IN LEFT SHIFT 04221000 D
0C8C  4809 E002 0000 00F0          NOT A3                                           04222000 D
0C8D  63C9 0000 0000 00F0          IF NOT ABT THEN SET LC1                          04223000 D
0C8E  5020 0000 0000 006C          OVBIT - 1 = CPCR        % SET THE OVERFLOW BIT   04224000 D
0C8F  4809 0C5E 0000 80F0          -B = SAR                % SHIFT AMOUNT            04225000 D
0C90  49C9 C001 1B70 C0F0          A2 L = A3,B; SET LC1    % PERFORM SHIFT          04226000 D
0C91  2000 0000 0000 006C          SETCCA - 1 = CPCR       % SET THE CONDITION BITS  04227000 D
0C92  4809 E001 2000 C0F0          A3 L = A2               % (R(A+1)) INTO MS WORD OF A2 04228000 D
0C93  0000 0000 0000 0020          COMP 16 = SAR                                    04229000 D
0C94  4809 C000 8030 00F0          A2 R = MIR              % (R(A+1)) INTO MIR       04230000 D
0C95  2F50 0000 0030 006C          EOUTPUT - 1 = CPCR                               04231000 D
0C96  4809 E000 8030 00F0          A3 R = MIR                                       04232000 D
0C97  4809 0000 0000 20F0          ASR                     % REFERENCE BR1          04233000 D
0C98  4809 0F40 801C C0F0          BMAR R = MAR2                                    04234000 D
0C99  0000 0000 0030 0010          8 = SAR                                          04235000 D
0C9A  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR                               04236000 D
0C9B  66E0 0000 C000 0040          OPCODE - 1 = MPCR                                04237000 D
                                                           %                        04238000 D
                                                           %                        04239000 D
                                OP163:                     % RX TYPE STORE AND INDEX BY 2 04240000 D
                                                           % (R(A),R(A+1)) INTO Y, Y+1 04241000 D
                                                           % (R(M)) + 2 INTO R(M)    04242000 D
0C9C  5700 0000 0C00 0060          RXMFIELD - 1 = CPCR     % Y INTO B                04243000 D
0C9D  4809 0C41 0020 00F0          B L = BR1               % TEMP STORAGE OF Y INTO BR1 04244000 D
0C9E  0000 0000 0030 0030          COMP 8 = SAR                                     04245000 D
0C9F  2380 0000 0030 0060          CONTENTSRM - 1 = CPCR   % (R(M)) INTO B          04246000 D
0CA0  4809 2C40 0080 00F0          LIT + B = MIR                                    04247000 D
0CA1  0020 0000 0030 00E0          2 = LIT                                          04248000 D
0CA2  2F50 0000 0030 006C          EOUTPUT - 1 = CPCR      % (R(M)) + 2 INTO R(M)   04249000 D
0CA3  4809 E000 9000 00F0          A3 R = A3                                        04250000 D
0CA4  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                04251000 D
0CA5  4809 E156 0B00 00F0          A3 AND LIT = B          % ISOLATE "A"            04252000 D
0CA6  51C0 0000 0000 0060          REGSTACK - 1 = CPCR     % ADDR OF R(A) INTO MAR2 04253000 D
0CA7  2F30 0000 0000 0060          EINPUT - 1 = CPCR       % (R(A)) INTO B          04254000 D
0CA8  4809 0C40 0030 00F0          B = MIR                                          04255000 D
0CA9  4809 2F5C 1030 C0F0          LIT OR BMAR = A3        % R(A+1)                 04256000 D
0CAA  001C 0000 0000 00E0          1 = LIT                                          04257000 D
```

```
0CAB  4809 00C0 0000 20F0                ASR                          % REFERENCE BR1                  04258000 D
0CAC  4809 0F43 001C 00F0                BMAR = BR2                   % ADDR OF Y INTO BR2             04259000 D
0CAD  61E0 0000 0090 0060                EMULOUT - 1 = CPCR           % (R(A)) INTO Y                  04260000 D
0CAE  4809 0F47 0020 00F0                BMAR + 1 L = BR1             % ADDR OF Y + 1                  04261000 D
0CAF  0000 00C0 0030 0030                COMP 8 = SAR                                                 04262000 D
0CB0  4809 E003 001C 00F0                A3 = MAR2                                                    04263000 D
0C91  2F30 0000 0000 006C                EINPUT - 1 = CPCR            % (R(A+1)) INTO B                04264000 D
0CB2  4809 0C40 0030 00F0                B = MIR                                                      04265000 D
0CB3  4809 0000 0000 20F0                ASR                          % REFERENCE BR1                  04266000 D
0CB4  4809 0F43 001C 00F0                BMAR = BR2                                                   04267000 D
0CB5  61E0 0000 0000 006C                EMULOUT - 1 = CPCR           % (R(A+1)) INTO Y+1              04268000 D
0CB6  66EC 00C0 0030 C040                OPCODE - 1 = MPCR                                            04269000 D
                                                                      %                               04270000 D
                                                                      %                               04271000 D
0CB7  5F9C 0000 0000 0060        OPCODE17:  XFCODE - 1 = CPCR         % "F" INTO A2                    04272000 D
0CB8  4809 C640 0030 00F0                A2 + AMPCR = AMPCR                                           04273000 D
0CB9  1640 0000 0000 00C0                OP17F - 1 = AMPCR                                            04274000 D
0CBA  4809 00C0 0000 00F0                STEP                                                         04275000 D
0CBB  4824 0000 0000 00F0                EXEC                                                         04276000 D
                                                                      %                               04277000 D
                                                                      %                               04278000 D
0CBC  1650 00C0 0000 004C        OP17F:  OP17C - 1 = MPCR                                             04279000 D
0CBD  166C 0000 0030 0040                OP171 - 1 = MPCR                                             04280000 D
0CBE  167C 00C0 0030 C040                OP172 - 1 = MPCR                                             04281000 D
0CBF  168C 00C0 0030 C040                OP173 - 1 = MPCR                                             04282000 D
                                                                      %                               04283000 D
                                                                      %                               04284000 D
                                 OP170:                               % RR TYPE CIRCULAR DOUBLE LEFT SHIFT 04285000 D
                                                                      % SHIFT (R(A),R(A+1)) LEFT CIRCULARLY 04286000 D
                                                                      % (R(M)) (0-5) PLACES AND SET CC 04287000 D
0CC0  2380 0000 0000 0060                CONTENTSRM - 1 = CPCR        % (R(M)) INTO B                  04288000 D
0CC1  4809 0C40 2000 00F0                B = A2                                                       04289000 D
0CC2  4809 E000 8800 00F0                A3 R = B                                                     04290000 D
0CC3  80F0 0000 0000 0080                4 = SAR; 15 = LIT                                            04291000 D
0CC4  4809 2C56 0090 00F0                LIT AND 6 = B                                                04292000 D
0CC5  51C0 00C0 0000 0060                REGSTACK - 1 = CPCR                                          04293000 D
0CC6  4809 0F41 0020 00F0                BMAR L = BR1                 % TEMP STORAGE OF R(A)          04294000 D
0CC7  001C 0000 0000 00B0                COMP 8 = SAR; 1 = LIT                                        04295000 D
0CC8  4809 2F5C 001C 00F0                LIT OR BMAR = MAR2           % R(A+1)                        04296000 D
0CC9  2F30 00C0 0050 C060                EINPUT - 1 = CPCR            % (R(A)) INTO B                  04297000 D
0CCA  4809 0F45 001C 00FC                BMAR + 1 = MAR2                                              04298000 D
0CCB  4809 CC41 1000 00FC                B L = A3                     % A3 HAS (R(A)) IN MS WORD       04299000 D
0CCC  0000 00C0 0000 0020                COMP 16 = SAR                                                04300000 D
0CCD  2F30 00C0 0090 0060                EINPUT - 1 = CPCR            % (R(A+1)) INTO B                04301000 D
0CCE  4809 EC5C 1030 00F0                A3 OR B = A3                 % A3 = (R(A))/(R(A+1))           04302000 D
0CCF  4809 C002 2000 00FC                NOT A2 = A2                                                  04303000 D
0CD0  4809 CCC0 C000 80F0                A2 + 1 = SAR                 % COMP OF (R(M)) (0-5)           04304000 D
0CD1  49C9 E001 9B30 00F0                A3 C = A3, B; SET LC1        % PERFORM SHIFT                  04305000 D
0CD2  20D0 0000 0030 0060                SETCCA - 1 = CPCR            % SET THE CONDITION BITS         04306000 D
0CD3  4809 E001 0B30 00F0                A3 L = B                     % (R(A+1)) INTO B                04307000 D
0CD4  0000 00C0 0030 0020                COMP 16 = SAR                                                04308000 D
0CD5  4809 0C40 8030 C0F0                B R = MIR                    % (R(A+1)) INTO MIR              04309000 D
0CD6  2F50 0000 0000 C060                EOUTPUT - 1 = CPCR           % WRITE SHIFTED (R(A+1))         04310000 D
0CD7  4809 00C0 0000 20F0                ASR                          % REFERENCE BR1                  04311000 D
0CD8  4809 0F43 001C 00F0                BMAR R = MAR2                                                04312000 D
0CD9  0000 00C0 0000 0010                B = SAR                                                      04313000 D
0CDA  4809 E000 8030 00F0                A3 R = MIR                                                   04314000 D
0CDB  0000 00C0 0030 002C                16 = SAR                                                     04315000 D
0CDC  2F50 00C0 0000 C060                EOUTPUT - 1 = CPCR                                           04316000 D
0CDD  66E0 0000 0000 004C                OPCODE - 1 = MPCR                                            04317000 D
```

```
                                                  %                                                  04319000 D
                                OP171:            % RI TYPE 1 STORE ZERO, 0 INTO Y*                   04320000 D
0CDE  2380 0000 0030 0060                CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                      04321000 D
0CDF  4809 0C41 0010 00F0                B L = BR2                                                   04322000 D
0CE0  0000 0000 0000 0030                COMP 8 = SAR                                                04323000 D
0CE1  4809 0000 0030 00F0                0 = MIR                                                     04324000 D
0CE2  61E0 0000 0000 0060                EMULOUT - 1 = CPCR                                          04325000 D
0CE3  66E0 0000 0000 0040                OPCODE - 1 = MPCR                                           04326000 D
                                                  %                                                  04327000 D
                                                  %                                                  04328000 D
                                OP172:            % RK TYPE CIRCULAR LEFT DOUBLE SHIFT                04329000 D
                                                  % SHIFT (R(A),R(A+1)) LEFT CIR. Y (0-5)             04330000 D
                                                  % AND SET CC                                        04331000 D
0CE4  4809 2056 0B30 00F0                LIT AND B = B                                               04332000 D
0CE5  00FF 0000 0000 0EE0                15 = LIT                                                    04333000 D
0CE6  4809 0C52 0030 00F0                C EQL B                                                     04334000 D
0CE7  6819 0000 0000 00F0                IF TRUE THEN SKIP                                           04335000 D
0CE8  2380 0000 0000 0060                CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                      04336000 D
0CE9  4809 0C41 0B30 00F0                B L = B                                                     04337000 D
0CEA  0000 0000 0030 C020                COMP 16 = SAR                                               04338000 D
0CEB  4809 EC5C 1030 00F0                A3 OR B = A3           % A3 = (R(M))/INSTRUCTION            04339000 D
0CEC  6330 0000 0000 0060                IFETCH - 1 = CPCR      % "Y" INTO B                         04340000 D
0CED  4809 EC00 A000 00F0                A3 R = A2              % (R(M)) INTO A2                      04341000 D
0CEE  0000 0000 0030 0020                16 = SAR                                                    04342000 D
0CEF  4809 CC40 2000 00F0                A2 + B = A2            % Y INTO A2                           04343000 D
0CF0  4809 EC00 9000 C0F0                A3 R = A3                                                   04344000 D
0CF1  90F0 0000 0000 0080                4 = SAR; 15 = LIT                                           04345000 D
0CF2  4809 E155 0B30 C0F0                A3 AND LIT = B         % ISOLATE "A"                        04346000 D
0CF3  5100 0000 0000 0060                REGSTACK - 1 = CPCR                                         04347000 D
0CF4  4809 0F41 0020 00F0                BMAR L = BR1           % TEMP STORAGE OF R(A)               04348000 D
0CF5  001C 0000 0000 C080                COMP 8 = SAR; 1 = LIT                                       04349000 D
0CF6  2F30 0000 0000 0060                EINPUT - 1 = CPCR      % (R(A)) INTO E                      04350000 D
0CF7  4809 2F5C 001C 00F0                LIT OR BMAR = MAR2     % R(A+1)                             04351000 D
0CF8  4809 0C41 1000 C0F0                P L = A3               % (R(A)) INTO PS WORD OF A3          04352000 D
0CF9  0000 0000 0030 0020                COMP 16 = SAR                                               04353000 D
0CFA  2F30 0000 0000 0060                EINPUT - 1 = CPCR      % (R(A+1)) INTO B                    04354000 D
0CFB  4809 EC5C 1000 00F0                A3 OR B = A3           % A3 = (R(A))/(R(A+1))               04355000 D
0CFC  4809 CC02 2000 00F0                NOT A2 = A2                                                 04356000 D
0CFD  4809 C0C0 0000 80F0                A2 + 1 = SAR           % COMP OF Y (0-5)                    04357000 D
0CFE  49C9 E001 9B30 C0F0                A3 C = A3,B; SET LC1   % PERFORM SHIFT                      04358000 D
0CFF  2000 0000 0000 0060                SETCCA - 1 = CPCR      % SET THE CONDITION BITS             04359000 D
0D00  4809 E001 2030 C0F0                A3 L = A2                                                   04360000 D
0D01  0000 0000 0030 0020                COMP 16 = SAR                                               04361000 D
0D02  4809 C0C0 8030 C0F0                A2 R = MIR             % SHIFTED (R(A+1)) INTO MIF          04362000 D
0D03  2F5C 0000 0000 0060                EOUTPUT - 1 = CPCR                                          04363000 D
0D04  4809 0000 0000 20F0                ASR                    % REFERENCE BR1                      04364000 D
0D05  4809 0F40 801C C0F0                BMAR R = MAR2                                               04365000 D
0D06  0000 0000 0000 C010                8 = SAR                                                     04366000 D
0D07  4809 E000 8030 00F0                A3 R = MIR                                                  04367000 D
0D08  0000 0000 0000 C020                16 = SAR                                                    04368000 D
0D09  2F50 0000 0000 0060                EOUTPUT - 1 = CPCR                                          04369000 D
0D0A  66E0 0000 0000 0040                OPCODE - 1 = MPCR                                           04370000 D
                                                  %                                                  04371000 D
                                                  %                                                  04372000 D
                                OP173:            % RX TYPE STORE ZERO, 0 INTO Y                      04373000 D
0D0B  57C0 0000 0030 0060                RXMFIELD - 1 = CPCR    % ADDR OF Y INTO B                   04374000 D
0D0C  4809 0C41 0010 00F0                B L = BR2                                                   04375000 D
0D0D  0000 0000 0000 0030                COMP 8 = SAR                                                04376000 D
0D0E  4809 0000 0030 00F0                0 = MIR                                                     04377000 D
```

```
0010  66E0 0000 0030 0040              OPCODE - 1 = MPCR                                  04379000 D
                                                                  %                       04380000 D
                                                                  %                       04381000 D
0011  5F9C 0000 0000 0060      OPCODE20:  XECODE - 1 = CPCR       % "F" INTO A2           04382000 D
0012  4809 C640 0040 00F0              A2 + AMPCR = AMPCR                                 04383000 D
0013  169C 0000 0030 00C0              OP20F - 1 = AMPCR                                  04384000 D
0014  4809 0000 0000 00F0              STEP                                               04385000 D
0015  4824 0000 0030 00F0              EXEC                                               04386000 D
                                                                  %                       04387000 D
0016  16A0 0000 0000 0040      OP20F:  OP200 - 1 = MPCR                                   04388000 D
0017  16B0 0000 0030 0040              OP201 - 1 = MPCR                                   04389000 D
0018  16C0 0000 0000 0040              OP202 - 1 = MPCR                                   04390000 D
0019  16D0 0000 0030 0040              OP203 - 1 = MPCR                                   04391000 D
                                                                  %                       04392000 D
                                                                  %                       04393000 D
                               OP200:                             % RR TYPE SUBTRACT REGISTER 04394000 D
                                                                  % (R(A)) - (R(M)) INTO R(A) 04395000 D
                                                                  % SET CC, CARRY AND OVERFLOW 04396000 D
001A  2280 0000 0000 0060              CONTENTSRA - 1 = CPCR  % (R(A)) INTO B             04397000 D
001B  2809 0C41 2030 00F0              B L = A2; IF LC1       % (R(A)) INTO MS WORD OF A2  04398000 D
001C  00F0 0000 0000 00A0              COMP 16 = SAR; 15 = LIT                            04399000 D
001D  4809 E155 0830 00F0              A3 AND LIT = B                                     04400000 D
001E  51C0 0000 0030 0060              REGSTACK - 1 = CPCR                                04401000 D
001F  2F30 0000 0000 0060              EINPUT - 1 = CPCR      % (R(M)) INTO B             04402000 D
0020  4849 0C41 0800 00F0              B L = B; SET LC2                                   04403000 D
0021  4809 CC5E 0030 00F0              A2 - B = MIR           % (R(A)) - (R(M)) INTO MIR  04404000 D
0022  7BC9 0000 0030 00F0              IF AOV THEN SET LC1                                04405000 D
0023  1E70 0000 0000 0060              CARRY - 1 = CPCR       % SET THE CARRY BIT         04406000 D
0024  4F10 0000 0030 0060              CHECKOV - 1 = CPCR     % SET THE OVERFLOW BIT      04407000 D
0025  4809 0000 0030 00F0              BMI                                                04408000 D
0026  4809 0C40 8830 00F0              B R = B,MIR                                        04409000 D
0027  00F0 0000 0000 0020              16 = SAR                                           04410000 D
0028  4809 E000 8010 00F0              A3 R = MAR2            % ADDR OF R(A)              04411000 D
0029  2F50 0000 0000 0060              EOUTPUT - 1 = CPCR                                 04412000 D
002A  2D00 0000 0000 0060              SETCCA - 1 = CPCR      % SET THE CONDITION BITS    04413000 D
002B  66E0 0000 0000 0040              OPCODE - 1 = MPCR                                  04414000 D
                                                                  %                       04415000 D
                                                                  %                       04416000 D
                               OP201:                             % RI TYPE 2 SUBTRACT (INDIRECT) 04417000 D
                                                                  % (R(A)) - (Y*) INTO R(A), 04418000 D
                                                                  % SET CC, CARRY, AND OV BITS 04419000 D
002C  2380 0000 0030 0060              CONTENTSRM - 1 = CPCR  % (R(M)) INTO B             04420000 D
002D  4809 0C41 0020 00F0              B L = BR1              % Y* INTO BR1                04421000 D
002E  0000 0000 0030 003C              COMP 8 = SAR                                       04422000 D
002F  4809 E000 0800 00F0              A3 = B                                             04423000 D
0030  2280 0000 0030 0060              CONTENTSRA - 1 = CPCR  % (R(A)) INTO B             04424000 D
0031  4809 0C41 2000 00F0              B L = A2               % (R(A)) INTO MS WORD OF A2  04425000 D
0032  0000 0000 0030 0020              COMP 16 = SAR                                      04426000 D
0033  4809 0000 0000 20F0              ASR                    % REF BR1                   04427000 D
0034  4809 0F40 0010 00F0              BMAR = BR2                                         04428000 D
0035  60E0 0000 0000 0060              EMULIN - 1 = CPCR      % (Y*) INTO B               04429000 D
0036  4809 0C41 0800 00F0              B L = B                % (Y*) INTO MS WORD OF B     04430000 D
0037  0000 0000 0030 002C              COMP 16 = SAR                                      04431000 D
0038  4849 CC5E 0030 00F0              A2 - B = MIR; SET LC2 % (R(A)) - (Y*) INTO MIR     04432000 D
0039  7BC9 0000 0030 00F0              IF AOV THEN SET LC1                                04433000 D
003A  1E70 0000 0030 0060              CARRY - 1 = CPCR                                   04434000 D
003B  4F10 0000 0000 0060              CHECKOV - 1 = CPCR                                 04435000 D
003C  4809 0000 0030 00F0              BMI                                                04436000 D
003D  4809 0C40 8830 00F0              B R = B, MIR                                       04437000 D
```

```
003E  0000 0000 0030 0020          16 = SAR                                   04438000 D
003F  4809 E003 801C 00F0          A3 R = MAR2              % ADDR OF R(A)    04439000 D
0040  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR       % WRITE OUT NEW R(A)  04440000 D
0041  2000 0000 0000 C06C          SETCCA - 1 = CPCR        % SET THE CONDITION BITS  04441000 D
0042  66E0 0000 0030 C040          OPCODE - 1 = MPCR                          04442000 D
                                                            %                  04443000 D
                                                            %                  04444000 D
                                OP202:                      %TYPE RK SUBTRACT, (R(A)) -Y INTO R(A)  04445000 D
                                                            % SET CC, CARRY AND OV BITS  04446000 D
0043  4809 2056 0800 00FC          LIT AND B = B            % "H" INTO B       04447000 D
0044  00FC 0000 0030 00EC          15 = LIT                                   04448000 D
0045  4809 CC52 0030 C0F0          0 EQL B                                    04449000 D
0046  6819 0000 0000 00F0          IF TRUE THEN SKIP                          04450000 D
0047  2380 0000 0000 0060          CONTENTSRH - 1 = CPCR  % (R(H)) INTO B     04451000 D
0048  4809 CC41 0B30 00F0          B L = B                                    04452000 D
0049  C000 0000 0000 0020          COMP 16 = SAR                              04453000 D
004A  4809 EC5C 1030 00FC          A3 OR B = A3             % A3 = (R(H))/INSTRUCTION  04454000 D
004B  633C 0000 0000 C06C          IFETCH - 1 = CPCR        % "Y" INTO B      04455000 D
004C  4809 E003 A000 00F0          A3 R = A2                % (R(H)) INTO A2   04456000 D
004D  0000 0000 0000 0020          16 = SAR                                   04457000 D
004E  4809 CC41 0030 00F0          A2 + B L = MIR           % Y INTO MS WORD OF MIR  04458000 D
004F  4809 E000 0B30 00F0          A3 = B                                     04459000 D
0050  2280 0000 0000 C060          CONTENTSRA - 1 = CPCR                      04460000 D
0051  4809 0C41 2000 00F0          B L = A2, BMI            % (R(A)) INTO MS WORD OF A2  04461000 D
0052  0000 0000 0030 002C          COMP 16 = SAR                              04462000 D
0053  4849 CC5E 0030 00F0          A2 - B = MIR; SET LC2  % (R(A)) - (Y) INTO MIR  04463000 D
0054  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                        04464000 D
0055  1E70 0000 0030 0060          CARRY - 1 = CPCR         % SET THE CARRY BIT  04465000 D
0056  9F10 0000 0000 0060          CHECKOV - 1 = CPCR       % SET THE OV BIT   04466000 D
0057  4809 0003 0000 00F0          BMI                                        04467000 D
0058  4809 0C40 8B30 00F0          B R = B, MIR                               04468000 D
0059  0000 0000 0000 0020          16 = SAR                                   04469000 D
005A  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR       % WRITE OUT NEW (R(A))  04470000 D
005B  2000 0000 0000 0060          SETCCA - 1 = CPCR        % SET THE CONDITION BITS  04471000 D
005C  66E0 0000 0000 0040          OPCODE - 1 = MPCR                          04472000 D
                                                            %                  04473000 D
                                                            %                  04474000 D
                                OP203:                      % TYPE RX SUBTRACT, (R(A))-(Y) -> R(A)  04475000 D
                                                            % SET CC, CARRY AND OV BITS  04476000 D
005D  5700 0000 0030 C060          RXMFIELD - 1 = CPCR      % ADDR OF Y INTO B  04477000 D
005E  4809 0C41 0010 00FC          B L = BR2                                  04478000 D
005F  0000 0000 0030 0030          COMP 8 = SAR                               04479000 D
0060  60EC 0000 0000 C06C          EMULIN - 1 = CPCR        % (Y) INTO B      04480000 D
0061  4809 0C41 0030 00F0          B L = MIR                % (Y) INTO MS WORD OF MIR  04481000 D
0062  0000 0000 0030 0020          COMP 16 = SAR                              04482000 D
0063  4809 E000 0B30 00F0          A3 = B                                     04483000 D
0064  2280 0000 0000 C060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B     04484000 D
0065  4809 0C41 2030 00F0          B L = A2, BMI            % (R(A)) INTO MS WORD OF A2  04485000 D
0066  0000 0000 0000 0020          COMP 16 = SAR                              04486000 D
0067  4849 CC5E 0030 00F0          A2 - B = MIR; SET LC2  % (R(A)) - (Y) INTO MIR  04487000 D
0068  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                        04488000 D
0069  1E70 0000 0030 0060          CARRY - 1 = CPCR                           04489000 D
006A  9F10 0000 0000 0060          CHECKOV - 1 = CPCR                         04490000 D
006B  4809 0003 0030 00F0          BMI                                        04491000 D
006C  4809 0C40 8B30 00F0          B R = B, MIR                               04492000 D
006D  0000 0000 0000 0020          16 = SAR                                   04493000 D
006E  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR       % WRITE NEW (R(A))  04494000 D
006F  2000 0000 0000 0060          SETCCA - 1 = CPCR        % SET THE CONDITION BITS  04495000 D
0070  66E0 0000 0000 0040          OPCODE - 1 = MPCR                          04496000 D
                                                            %                  04497000 D
```

```
                                                                  %                                                  04498000 D
0071  5F90 0000 0000 0060   OPCODE21:  XFCODE - 1 = CPCR          % "F" INTO A2                                      04499000 D
0072  4809 C640 0040 00FC              A2 + AMPCR = AMPCR                                                            04500000 D
0073  16EC 0000 0030 00C0              OP21F - 1 = AMPCR                                                             04501000 D
0074  4809 03C0 0000 00F0              STEP                                                                          04502000 D
0075  4824 0000 0030 C0F0              EXEC                                                                          04503000 D
                                                                  %                                                  04504000 D
0076  16FC 0000 0030 0040   OP21F:     OP210 - 1 = MPCR                                                              04505000 D
0077  1700 0000 0000 0040              OP211 - 1 = MPCR                                                              04506000 D
0078  3000 0000 0030 0040              FAULT - 1 = MPCR                                                              04507000 D
0079  171C 0000 0000 004C              OP213 - 1 = MPCR                                                              04508000 D
                                                                  %                                                  04509000 D
                                                                  %                                                  04510000 D
                            OP210:                                % TYPE RR SUBTRACT DOUBLE (REGISTER)                04511000 D
                                                                  % (R(A),R(A+1))-(R(M),R(M+1)) INTO                 04512000 D
                                                                  % R(A),R(A+1), SET CC, OV, AND CARRY                04513000 D
007A  4809 0C40 8800 00F0              B R = B                    % ISOLATE "A"                                       04514000 D
007B  30F0 0000 0030 0080              4 = SAR; 15 = LIT                                                             04515000 D
007C  4809 2C56 0800 00F0              LIT AND B = B                                                                 04516000 D
007D  51CC 0000 0030 0060              REGSTACK - 1 = CPCR                                                           04517000 D
007E  2F3C 0000 0000 006C              EINPUT - 1 = CPCR          % (R(A)) INTO B                                   04518000 D
007F  4809 0C41 2030 C0F0              B L = A2                   % (R(A)) INTO MS WORD OF A2                        04519000 D
0080  0000 0000 0000 C020              COMP 16 = SAR                                                                 04520000 D
0081  4809 0F41 0020 00FC              BMAR L = BR1               % TEMP STORAGE OF ADDR OF R(A)                     04521000 D
0082  001C 00C0 0030 00B0              COMP 8 = SAR; 1 = LIT                                                         04522000 D
0083  4809 2F5C 001C 00F0              LIT OR BMAR = MAR2         % R(A+1)                                           04523000 D
0084  2F3C 0000 0000 0060              EINPUT - 1 = CPCR          % (R(A+1)) INTO B                                  04524000 D
0085  4809 CC5C 2000 00F0              A2 OR B = A2               % A2 = (R(A))/(R(A+1))                             04525000 D
0086  4809 E156 0030 00F0              A3 AND LIT = B             % ISOLATE "M"                                      04526000 D
0087  00F0 0000 0030 00EC              15 = LIT                                                                      04527000 D
0088  51C0 0000 0000 0060              REGSTACK - 1 = CPCR                                                           04528000 D
0089  2F3C 0000 0000 0060              EINPUT - 1 = CPCR          % (R(M)) INTO B                                   04529000 D
008A  4809 0C41 1030 00F0              B L = A3                   % (R(M)) INTO MS WORD OF A3                        04530000 D
008B  0010 0000 0000 00A0              COMP 16 = SAR; 1 = LIT                                                        04531000 D
008C  4809 2F5C 001C 00F0              LIT OR BMAR = MAR2         % ADDR OF R(M+1)                                   04532000 D
008D  2F30 0000 0000 0060              EINPUT - 1 = CPCR          % (R(M+1)) INTO B                                  04533000 D
008E  4809 EC5C 0B30 00F0              A3 OR B = B                % B = (R(M))/(R(M+1))                              04534000 D
008F  4849 CC5E 0080 00F0              A2 - D = MIR; SET LC2      % PERFORM SUBTRACTION                              04535000 D
0090  7BC9 0000 0000 00F0              IF AOV THEN SET LC1                                                           04536000 D
0091  1E70 0000 0000 0060              CARRY - 1 = CPCR                                                              04537000 D
0092  4F10 0000 0000 0060              CHECKOV - 1 = CPCR                                                            04538000 D
0093  49C9 0000 0000 00F0              BMI; SET LC1                                                                  04539000 D
0094  2000 0000 0000 0060              SETCCA - 1 = CPCR          % SET THE CONDITION BITS                           04540000 D
0095  4809 0000 0030 00F0              BMI                                                                           04541000 D
0096  4809 0C40 8C90 C0F0              B R = MIR                  % NEW (R(A)) INTO MIR                              04542000 D
0097  0000 0000 0030 002C              16 = SAR                                                                      04543000 D
0098  4809 0C41 0800 C0F0              B L = B                                                                       04544000 D
0099  4809 0C40 8800 00F0              B R = B                    % (R(A+1))                                         04545000 D
009A  4809 00C0 0030 20F0              ASR                        % REFERENCE BR1                                    04546000 D
009B  4809 0F40 9010 00F0              BMAR R = MAR2, A3          % ADDR OF R(A)                                     04547000 D
009C  000C 0000 0000 001C              8 = SAR                                                                       04548000 D
009D  2F5C 0000 0030 0060              EOUTPUT - 1 = CPCR         % WRITE NEW (R(A))                                 04549000 D
009E  4809 EP0C 001C 00F0              A3 OR 1 = MAR2             % ADDR OF R(A+1)                                   04550000 D
009F  4809 0C40 0030 00F0              B = MIR                    % (R(A+1))                                         04551000 D
00A0  2F50 0000 0000 0060              EOUTPUT - 1 = CPCR         % WRITE NEW (R(A+1))                               04552000 D
00A1  66EC 0000 0030 0040              OPCODE - 1 = MPCR                                                             04553000 D
                                                                  %                                                  04554000 D
                                                                  %                                                  04555000 D
                            OP211:                                % RI TYPE 2 SUBTRACT DOUBLE                         04556000 D
                                                                  % (R(A),R(A+1))-(Y*,Y*+1) INTO                      04557000 D
```

```
                                                                  %R(A),R(A+1), SET CC, CARRY AND OV BITS   04558C00 D
0DA2  4809 0C40 8030 00F0          B R = B                                                                  04559C00 D
0DA3  80F0 00C0 0030 C080          4 = SAR; 15 = LIT                                                        04560C0C D
0DA4  4809 2C55 0030 00F0          LIT AND B = B                   % ISOLATE "A"                            04561C00 D
0DA5  51C0 00C0 0030 C06C          REGSTACK - 1 = CPCR                                                      04562C00 D
0DA6  2F30 00C0 0000 C06C          EINPUT - 1 = CPCR               % (R(A)) INTO E                          04563C0C D
0DA7  4809 0C41 2030 00F0          B L = A2                        % (R(A)) INTO MS WORD OF A2               04564C00 D
0DA8  000C 00C0 0030 C020          COMP 16 = SAR                                                            04565C0C D
0DA9  4809 0F41 0C20 00F0          BMAR L = BR1                    % SAVE ADDR OF R(A)                       04566C00 D
0DAA  0010 00C0 0000 C0B0          COMP 8 = SAR; 1 = LIT                                                    04567C0C D
0DAB  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2              % R(A+1)                                 04568C0C D
0DAC  2F30 0000 0000 C06C          EINPUT - 1 = CPCR               % (R(A+1)) INTO B                        04569C00 D
0DAD  4809 CC5C 2030 00F0          A2 OR B = A2                    % A2 = (R(A))/(R(A+1))                    04570C00 D
0DAE  4809 E156 0B00 00F0          A3 AND LIT = B                  % ISOLATE "M"                             04571C00 D
0DAF  00FC 0000 0030 C0E0          15 = LIT                                                                 04572C00 D
0DB0  51C0 0000 0030 006C          REGSTACK - 1 = CPCR              % (R(M)) INTO B                          04573C00 D
0DB1  2F30 0000 0000 0060          EINPUT - 1 = CPCR               % (R(M)) INTO B                          04574C00 D
0DB2  4809 0C41 0C10 00FC          B L = BR2                                                                04575C00 D
0DB3  000C 0000 0030 003C          COMP 8 = SAR                                                             04576C00 D
0DB4  60E0 00C0 0030 006C          EMULIN - 1 = CPCR               % (Y*) INTO B                            04577C00 D
0DB5  4809 2F5D 001C 00F0          LIT OR BMAR L = BR2             % ADDR OF Y*+1                           04578C00 D
0DB6  0010 00C0 0030 00B0          COMP 8 = SAR; 1 = LIT                                                    04579C00 D
0DB7  4809 0C41 1030 00FC          B L = A3                        % (Y*) INTO MS WORD OF A3                04580C0C D
0DB8  000C 0000 0030 0020          COMP 16 = SAR                                                            04581C0C D
0DB9  60E0 0000 0030 0060          EMULIN - 1 = CPCR               % (Y*+1) INTO C                          04582C0C D
0DBA  4809 EC5C 0B00 00F0          A3 OR B = B                     % B = (Y*)/(Y*+1)                         04583C00 D
0DBB  4849 CC5E 0C90 00F0          A2 - B = MIR; SET LC2            % PERFORM SUBTRACTION                    04584C0C D
0DBC  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                                                      04585C00 D
0DBD  1E70 0000 0000 0C60          CARRY - 1 = CPCR                                                         04586C00 D
0DBE  4F1C 0000 0030 0060          CHECKOV - 1 = CPCR              % SET THE OV BITS                        04587C0C D
0DBF  49C9 0000 0000 00F0          BMI; SET LC1                                                             04588C00 D
0DC0  2000 0000 0030 0060          SETCCA - 1 = CPCR               % SET THE CONDITION BITS                 04589C00 D
0DC1  4809 0000 0030 00FC          BMI                                                                      04590C00 D
0DC2  4809 0C40 8030 00F0          B R = MIR                       % NEW (R(A))                             04591C00 D
0DC3  0000 0000 0000 0020          16 = SAR                                                                 04592C00 D
0DC4  4809 0C41 0B30 00F0          B L = B                                                                  04593C00 D
0DC5  4809 0C40 8B30 00F0          B R = B                         % NEW (R(A+1))                           04594C00 D
0DC6  4809 0000 0030 20FC          ASR                             % REF BR1                                04595C00 D
0DC7  4809 0F40 8C1C 00F0          BMAR R = MAR2                   % ADDR OF R(A)                           04596C00 D
0DC8  001C 0000 0C00 C09C          8 = SAR; 1 = LIT                                                         04597C00 D
0DC9  2F50 00C0 0000 0060          EOUTPUT - 1 = CPCR              % WRITE NEW (R(A))                       04598C00 D
0DCA  4809 2F5C 001C 00FC          LIT OR BMAR = MAR2              % R(A+1)                                 04599C00 D
0DCB  4809 0C40 0030 00F0          B = MIR                         % (R(A+1))                               04600C00 D
0DCC  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR              % WRITE NEW (R(A+1))                     04601C00 D
0DCD  66EC 00C0 C0?C 0040          OPCODE - 1 = MPCR                                                        04602C00 D
                                                                   %                                        04603C00 D
                                                                   %                                        04604C0C D
                                   OP213:                          % RX TYPE SUBTRACT DOUBLE                 04605C00 D
                                                                   % (R(A),R(A+1))-(Y,Y+1)=R(A),R(A+1)       04606C00 D
                                                                   % SET CC, OV AND CARRY BITS              04607C00 D
0DCE  5700 00C0 0030 0060          RXMFIELD - 1 = CPCR             % Y ADDR INTO B                          04608C0C D
0DCF  4809 0C41 0010 00FC          B L = BR2                                                                04609C00 D
0DD0  000C 00C0 0030 0030          COMP 8 = SAR                                                             04610C00 D
0DD1  60EC 00C0 0C3C C06C          EMULIN - 1 = CPCR               % (Y) INTO B                             04611C00 D
0DD2  4809 0C41 2030 00F0          B L = A2                        % (Y) INTO MS WORD OF A2                  04612C00 D
0DD3  000C 0000 0030 Q02C          COMP 16 = SAR                                                            04613C00 D
0DD4  4809 2F5D 001C C0F0          LIT OR BMAR L = ER2             % ADDR OF Y+1                            04614C00 D
0DD5  0010 0000 0030 00BC          COMP 8 = SAR ; 1 = LIT                                                   04615C00 D
0DD6  60E0 00C0 0C00 C060          EMULIN - 1 = CPCR               % (Y+1) INTO B                           04616C0C D
0DD7  4809 CC5C 0090 00FC          A2 OR B = MIR                   % MIR = (Y)/(Y+1)                        04617C00 D
```

```
0FD8  4809 E0C0 0B00 C0FC          A3 = B
0FD9  4609 0C40 8B30 C0F0          B R = B                                          04619C00 D
0FDA  80F0 00C0 0000 0080          Q = SAR; 15 = LIT                                04620C00 D
0FDB  4809 2C55 0B0C C0F0          LIT AND B = 6                                    04621C00 D
0FDC  51C0 0000 C000 006C          REGSTACK - 1 = CPCR                              04622C00 D
0FDD  2F30 0000 0000 C06C          EINPUT - 1 = CPCR        % (R(A)) INTO B         04623C00 D
0FDE  4809 0F91 0020 C0F0          BMAR L = BR1             % SAVE ADDR OF R(A)      04624C00 D
0FDF  0000 00C0 0000 C030          COMP B = SAR                                     04625C00 D
0FE0  4809 0C41 2000 00FC          B L = A2                 % (R(A)) INTO MS WORD OF A2 04626000 D
0FE1  001C 0000 0000 C0A0          COMP 16 = SAR; 1 = LIT                           04627C00 D
0FE2  4809 2F5C 001C C0FC          LIT OR BMAR = MAR2       % R(A+1)                04628C00 D
0FE3  2F30 0000 0000 0060          EINPUT - 1 = CPCR        % (R(A+1)) INTO B       04629C00 D
0FE4  4809 CC5C 2030 00F0          A2 OR B = A2; BMI        % A2 = (R(A))/(R(A+1))   04630C0C D
0FE5  4849 CC5E 0080 C0FC          A2 - B = MIR; SET LC2  % PERFORM SUBTRACTION     04631C00 D
0FE6  7BC9 6000 0030 00FC          IF AOV THEN SET LC1                              04632C00 D
0FE7  1E7C 00C0 0030 C060          CARRY - 1 = CPCR                                 04633C00 D
0FE8  4F1C 0000 0000 0060          CHECKOV - 1 = CPCR                               04634C00 D
0FE9  49C9 00C0 0000 00F0          BMI; SET LC1                                     04635C00 D
0FEA  2000 0000 0000 0060          SETCCA - 1 = CPCR        % SET THE CONDITION BITS 04636C00 D
0FEB  4809 0000 0030 C0F0          BMI                                              04637C00 D
0FEC  4809 0C40 8030 00F0          B R = MIR                % NEW (R(A))             04638C00 D
0FED  0000 00C0 0030 C020          16 = SAR                                         04639C00 D
0FEE  4809 0C41 0B70 C0FC          B L = B                                          04640C00 D
0FEF  4809 CC40 8B30 00F0          B R = B                                          04641C00 D
0FF0  4809 00C0 0000 20F0          ASR                      % REF BR1               04642C00 D
0FF1  4809 0F40 901C C0FC          BMAR R = MAR2  ; A3                              04643C00 D
0FF2  000C 0000 C030 0010          B = SAR                                          04644C00 D
0FF3  2F50 00C0 0030 006C          EOUTPUT - 1 = CPCR       % WRITE NEW (R(A))      04645C00 D
0FF4  4809 E0DC 001C C0FC          A3 OR 1 = MAR2           % ADDR OF R(A+1)        04646C00 D
0FF5  4809 0C40 0C8C 00F0          B = MIR                                          04647C00 D
0FF6  2F50 00C0 0030 006C          EOUTPUT - 1 = CPCR       % WRITE NEW (R(A+1))    04648C00 D
0FF7  66E0 0000 0000 C040          OPCODE - 1 = MPCR                                04649C00 D
                                                            %                        04650C0C D
                                                            %                        04651C00 D
0FF8  5F9C 00C0 0000 C060   OPCODE22:  XFCODE - 1 = CPCR    % "F" INTO A2            04652C00 D
0FF9  4809 C640 0C40 C0F0          A2 + AMPCR = AMPCR                               04653C00 D
0FFA  1720 0000 0000 00CC          OP22F - 1 = AMPCR                                04654C00 D
0FFB  4809 0000 0C0C C0F0          STEP                                             04655C00 D
0FFC  4824 0000 0000 00F0          EXEC                                             04656C00 D
                                                            %                        04657C00 D
0FFD  173C 0000 0030 C040   OP22F:  OP220 - 1 = MPCR                                04658C00 D
0FFE  174C 00C0 0030 0040          OP221 - 1 = MPCR                                 04659C00 D
0FFF  175C 0000 0030 C040          OP222 - 1 = MPCR                                 04660C00 D
0E00  176C 0000 0030 0040          OP223 - 1 = MPCR                                 04661C00 D
                                                            %                        04662C00 D
                                                            %                        04663C00 D
                            OP220:                          % TYPE RR ADD, (R(A)) + (R(M)) -> R(A)  04664C00 D
                                                            % SET CC, CARRY AND OV BITS  04665C00 D
0E01  22B0 00C0 0000 C060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B          04666C00 D
0E02  4809 0C41 2030 00F0          B L = A2                                         04667C00 D
0E03  00FC 0000 0000 00A0          COMP 16 = SAR; 15 = LIT                          04668000 D
0E04  4809 E156 0B0C 00FC          A3 AND LIT = B                                   04669C00 D
0E05  51C0 0000 0030 C060          REGSTACK - 1 = CPCR      % ADDR OF R(M) INTO MAR2 04670C00 D
0E06  2F30 00C0 0030 C060          EINPUT - 1 = CPCR        % (R(M)) INTO B         04671000 D
0E07  4809 0C41 0B30 00FC          B L = B                                          04672C00 D
0E08  000C 00C0 0030 0020          COMP 16 = SAR                                    04673C00 D
0E09  4809 CC40 0B30 00F0          A2 + B = MIR             % PERFORM ADDITION       04674C00 D
0E0A  7BC9 0000 0030 00F0          IF AOV THEN SET LC1                              04675C00 D
0E0B  1E7C 00C0 0000 0060          CARRY - 1 = CPCR                                 04676C00 D
0E0C  4F1C 00C0 0030 C060          CHECKOV - 1 = CPCR                               04677C00 D
```

```
0E0D  4809 2C56 0B0C 00F0          A3 R = MARB2, BMI          % ADDR OF R(A)                04679000 D
0E0E  0000 0000 0000 0020          16 = SAR                                                 04680000 D
0E0F  4809 0C40 8830 C0FC          B R = MIR, B                                             04681000 D
0E10  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                       04682000 D
0E11  2000 0000 0000 0060          SETCCA - 1 = CPCR          % SET THE CONDITION BITS      04683000 D
0E12  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                        04684000 D
                                                              %                            04685000 D
                                                              %                            04686000 D
                                OP221:                        % TYPE RI TYPE 2 ADD, (R(A))+(Y*) ->  04687000 D
                                                              % R(A), SET CC, CARRY, AND OV BITS  04688000 D
0E13  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR      % (R(A)) INTO B               04689000 D
0E14  4809 0C41 2C00 00FC          B L = A2                   % (R(A)) INTO MS WORD OF A2   04690000 D
0E15  00F0 0000 0000 00A0          COMP 16 = SAR; 15 = LIT                                  04691000 D
0E16  4809 E156 0B00 00F0          A3 AND LIT = B                                           04692000 D
0E17  51C0 0000 0000 C06C          REGSTACK - 1 = CPCR                                      04693000 D
0E18  2F30 0000 0000 0060          EINPUT - 1 = CPCR          % (R(M)) INTO B              04694000 D
0E19  4809 0C41 0010 00FC          B L = BR2                                                04695000 D
0E1A  0000 0000 0000 0030          COMP B = SAR                                             04696000 D
0E1B  60E0 0000 0000 006C          EMULIN - 1 = CPCR          % (Y*) INTO B                04697000 D
0E1C  4809 0C41 0B0C 00F0          B L = B                    % (Y*) INTO MS WORD OF B      04698000 D
0E1D  0000 0000 0000 0020          COMP 16 = SAR                                            04699000 D
0E1E  4809 0C40 0C90 00F0          A2 + B = MIR                                             04700000 D
0E1F  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                                      04701000 D
0E20  1E70 0000 0000 0060          CARRY - 1 = CPCR                                         04702000 D
0E21  4F10 0000 0000 0060          CHECKOV - 1 = CPCR                                       04703000 D
0E22  4809 E0C3 8010 C0F0          A3 R = MAR2, BMI           % ADDR OF R(A)               04704000 D
0E23  0000 0000 0000 0020          16 = SAR                                                 04705000 D
0E24  4809 0C40 8830 C0F0          B R = MIR, B                                             04706000 D
0E25  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR         % WRITE NEW (R(A))           04707000 D
0E26  2000 0000 0000 0060          SETCCA - 1 = CPCR                                        04708000 D
0E27  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                        04709000 D
                                                              %                            04710000 D
                                                              %                            04711000 D
                                OP222:                        % TYPE RK ADD, (R(A))+Y -> R(A)  04712000 D
                                                              % SET CC, CARRY AND OV BITS   04713000 D
0E28  4809 2C56 0B0C 00F0          LIT AND B = B                                            04714000 D
0E29  00F0 0000 0000 00E0          15 = LIT                                                 04715000 D
0E2A  4809 0052 0000 00F0          0 EQL B                                                  04716000 D
0E2B  5819 0000 0000 00FC          IF TRUE THEN SKIP                                        04717000 D
0E2C  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR      % (R(M)) INTO B              04718000 D
0E2D  4809 0C41 0B00 00FC          B L = B                    % (R(M)) INTO MS WORD OF B   04719000 D
0E2E  0000 0000 0000 0020          COMP 16 = SAR                                            04720000 D
0E2F  4809 E05C 1000 00F0          A3 OR B = A3               % A3 = (R(M))/INSTRUCTION     04721000 D
0E30  6330 0000 0000 0060          IFETCH - 1 = CPCR          % "Y" INTO B                  04722000 D
0E31  4809 E000 A000 00FC          A3 R = A2                  % (R(M)) INTO A2              04723000 D
0E32  0000 0000 0000 0020          16 = SAR                                                 04724000 D
0E33  4809 0C41 0090 C0F0          A2 + B L = MIR             % Y INTO MIR                  04725000 D
0E34  4809 E0C3 0B00 00F0          A3 = B                                                   04726000 D
0E35  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR      % (R(A)) INTO B              04727000 D
0E36  4809 0C41 2000 00F0          B L = A2, BMI              % (R(A)) INTO MS WORD OF A2   04728000 D
0E37  0000 0000 0000 0020          COMP 16 = SAR                                            04729000 D
0E38  4809 0C40 0090 00F0          A2 + B = MIR               % PERFORM ADDITION            04730000 D
0E39  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                                      04731000 D
0E3A  1E70 0000 0000 0060          CARRY - 1 = CPCR                                         04732000 D
0E3B  4F10 0000 0000 0060          CHECKOV - 1 = CPCR                                       04733000 D
0E3C  4809 0000 0000 00F0          BMI                                                      04734000 D
0E3D  4809 0C40 8830 C0F0          B R = B, MIR                                             04735000 D
0E3E  0000 0000 0000 0020          16 = SAR                                                 04736000 D
0E3F  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                       04737000 D
0E40  2000 0000 0000 0060          SETCCA - 1 = CPCR          % SET THE CONDITION BITS     04738000 D
```

```
0E41  56E0 0000 0000 0040          OPCODE - 1 = MPCR                                  04738000 D
                                                          %                                04739000 D
                                                          %                                04740000 D
                                   OP223:                 % RX TYPE ADD, (R(A))+(Y) -> R(A) 04741000 D
                                                          % SET CC, CARRY, AND OV BITS    04742000 D
0E42  5700 0000 0030 0060          RXMFIELD - 1 = CPCR    % Y INTO B                     04743000 D
0E43  4809 0C41 0010 00F0          B L = BR2                                             04744000 D
0E44  0000 0000 0000 003C          COMP 8 = SAR                                          04745000 D
0E45  50E0 0000 0000 0060          EMULIN - 1 = CPCR      % (Y) INTO B                   04746000 D
0E46  4809 0C41 0090 00F0          B L = MIR              % (Y) INTO MS WORD OF MIR      04747000 D
0E47  0000 0000 0030 0020          16 = SAR                                              04748000 D
0E48  4809 E000 0B00 00F0          A3 = B                                                04749000 D
0E49  22B0 0000 0030 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                04750000 D
0E4A  4809 0C41 2000 00F0          B L = A2, BMI                                         04751000 D
0E4B  0000 0000 0000 0020          COMP 16 = SAR                                         04752000 D
0E4C  4809 C040 0030 00F0          A2 + B = MIR           % PERFORM ADDITION             04753000 D
0E4D  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                                   04754000 D
0E4E  1E70 0000 0000 0060          CARRY - 1 = CPCR                                      04755000 D
0E4F  4F10 0000 0030 0060          CHECKOV - 1 = CPCR                                    04756000 D
0E50  4809 0000 0000 00F0          BMI                                                   04757000 D
0E51  4809 0C40 8B30 00F0          B R = MIR, B                                          04758000 D
0E52  0000 0000 0030 0020          16 = SAR                                              04759000 D
0E53  2000 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS       04760000 D
0E54  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                    04761000 D
0E55  56E0 0000 0000 0040          OPCODE - 1 = MPCR                                     04762000 D
                                                          %                                04763000 D
                                                          %                                04764000 D
0E56  5F90 0000 0000 0060          OPCODE23:  XFCODE - 1 = CPCR  % "F" INTO A2           04765000 D
0E57  4809 C640 0030 00F0          A2 + AMPCR = AMPCR                                    04766000 D
0E58  1770 0000 0000 00D0          OP23F - 1 = AMPCR                                     04767000 D
0E59  4809 0000 0030 00F0          STEP                                                  04768000 D
0E5A  4824 0000 0000 00F0          EXEC                                                  04769000 D
                                                          %                                04770000 D
0E5B  1780 0000 0000 0040          OP23F:  OP230 - 1 = MPCR                              04771000 D
0E5C  179C 0000 0000 0040          OP231 - 1 = MPCR                                      04772000 D
0E5D  300C 0000 0000 0040          FAULT - 1 = MPCR                                      04773000 D
0E5E  17A0 0000 0000 0040          OP233 - 1 = MPCR                                      04774000 D
                                                          %                                04775000 D
                                                          %                                04776000 D
                                   OP230:                 % TYPE RR ADD DOUBLE, (R(A)),R(A+1)) + 04777000 D
                                                          % (R(M),R(M+1)) INTO R(A),R(A+1) 04778000 D
                                                          % SET CC, CARRY AND OV BITS     04779000 D
0E5F  4809 2056 0B30 00F0          LIT AND B = B          % ISOLATE "M"                   04780000 D
0E60  00F0 0000 0030 00E0          15 = LIT                                              04781000 D
0E61  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                                   04782000 D
0E62  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(M)) INTO B                04783000 D
0E63  4809 0C41 2030 00F0          B L = A2               % (R(M)) INTO MS WORD OF A2     04784000 D
0E64  0010 0000 0030 00A0          COMP 16 = SAR ; 1 = LIT                               04785000 D
0E65  4809 2F50 0010 00F0          LIT OR BMAR = MAR2     % ADDR OF R(M+1)               04786000 D
0E66  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(M+1)) INTO B              04787000 D
0E67  4809 0C50 0030 00F0          A2 OR B = MIR          % MIR = (R(M))/(R(M+1))         04788000 D
0E68  4809 E000 8B30 00F0          A3 R = B                                              04789000 D
0E69  30F0 0000 0000 00B0          4 = SAR; 15 = LIT                                     04790000 D
0E6A  4809 2056 0B00 00F0          LIT AND B = B          % ISOLATE "A"                   04791000 D
0E6B  51C0 0000 0070 0060          REGSTACK - 1 = CPCR                                   04792000 D
0E6C  4809 0F41 0020 00F0          BMAR L = BR1           % TEMP STORAGE                 04793000 D
0E6D  0000 0000 0000 003C          COMP 8 = SAR                                          04794000 D
0E6E  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B                04795000 D
0E6F  4809 0C41 2030 00F0          B L = A2               % (R(A)) INTO MS WORD OF A2     04796000 D
0E70  0010 0000 0000 00A0          COMP 16 = SAR ; 1 = LIT                               04797000 D
```

```
0E71  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2      % R(A+1)
0E72  2F30 0000 0000 0060          EINPUT - 1 = CPCR       % (R(A+1)) INTO B                    04799C00 D
0E73  4809 CC5C 2000 C0F0          A2 OR B = A2, BMI       % A2 = (R(A))/(R(A+1))               04800C00 D
0E74  4809 CC40 0030 C0F0          A2 + B = MIR            % PERFORM ADDITION                   04801000 D
0E75  7BC9 0000 0030 C0FC          IF AOV THEN SET LC1                                         04802C00 D
0E76  1E70 0000 0030 006C          CARRY - 1 = CPCR                                            04803C00 D
0E77  9F1C 0000 0000 0060          CHECKOV - 1 = CPCR                                          04804C00 D
0E78  49C9 0000 0000 C0F0          BMI; SET LC1                                                04805C00 D
0E79  2000 0000 0000 0060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS            04806C00 D
0E7A  4809 0000 0030 C0F0          BMI                                                         04807C00 D
0E7B  4809 0C41 2000 C0F0          B L = A2                                                    04808C00 D
0E7C  0000 0000 0030 C020          COMP 16 = SAR                                               04809C00 D
0E7D  4809 C000 8030 00F0          A2 R = MIR              % (R(A+1))                          04810C00 D
0E7E  4809 0C40 8800 00FC          B R = B                 % (R(A))                            04811C00 D
0E7F  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR      % WRITE NEW (R(A+1))                04812C00 D
0E80  4809 0000 0000 20FC          ASR                     % REF BR1                           04813C00 D
0E81  4809 0F43 801C 00FC          BMAR R = MAR2           % R(A)                              04814C00 D
0E82  0000 0000 0030 001C          B = SAR                                                     04815C00 D
0E83  4809 CC40 0030 00F0          B = MIR                                                     04816C00 D
0E84  2F5C 0000 0030 0060          EOUTPUT - 1 = CPCR      % WRITE NEW R(A)                    04817C00 D
0E85  56E0 0000 0030 0040          OPCODE - 1 = MPCR                                           04818C00 D
                                                           %                                   04819C00 D
                                                           %                                   04820C00 D
                                OP231:                     % RI TYPE 2 ADD DOUBLE               04821C00 D
                                                           % (R(A),R(A+1))+(Y*,Y*+1) INTO       04822C00 D
                                                           % R(A),R(A+1), SET CC, CARRY AND OV BITS 04823C00 D
0E86  2380 0000 0030 006C          CONTENTSRM - 1 = CPCR                                       04824C00 D
0E87  4809 0C41 001C 00F0          B L = BR2               % Y* INTO BR2                       04825C00 D
0E88  0000 0000 0030 0030          COMP 8 = SAR                                                04826C00 D
0E89  60EC 0000 0000 0060          EMULIN - 1 = CPCR       % (Y*) INTO B                       04827C00 D
0E8A  4809 0C41 2000 00F0          B L = A2                % (Y*) INTO MS WORD OF A2            04828C00 D
0E8B  0000 0000 0030 0020          COMP 16 = SAR                                               04829C00 D
0E8C  4809 0F47 0010 00FC          BMAR + 1 L = BR2                                            04830C00 D
0E8D  0000 0000 0000 C030          COMP 8 = SAR                                                04831C00 D
0E8E  60EC 0000 0030 006C          EMULIN - 1 = CPCR       % (Y*+1) INTO B                     04832C00 D
0E8F  4809 CC5C 0030 00F0          A2 OR B = MIR           % MIR = (Y*)/(Y*+1)                 04833C00 D
0E90  4809 E0C0 9030 00F0          A3 R = A3                                                   04834C00 D
0E91  8CFC 0000 0000 C080          4 = SAR; 15 = LIT                                           04835C00 D
0E92  4809 E15C 0030 00F0          A3 AND LIT = B                                              04836C00 D
0E93  51C0 0000 0030 0060          REGSTACK - 1 = CPCR                                         04837C00 D
0E94  4809 0F41 002C 00FC          BMAR L = BR1            % TEMP STORAGE FOR R(A)             04838C00 D
0E95  0000 0000 0000 003C          COMP 8 = SAR                                                04839C00 D
0E96  2F30 0000 0000 0060          EINPUT - 1 = CPCR       % (R(A)) INTO B                     04840C00 D
0E97  4809 0C41 2000 00F0          B L = A2                % (R(A)) INTO MS WORD OF A2          04841C00 D
0E98  0010 0000 0000 C0A0          COMP 16 = SAR; 1 = LIT                                      04842C00 D
0E99  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2      % R(A+1)                            04843C00 D
0E9A  2F30 0000 0000 0060          EINPUT - 1 = CPCR       % (R(A+1)) INTO B                   04844C00 D
0E9B  4809 CC5C 2000 C0F0          A2 OR B = A2, BMI       % A2 = (R(A))/(R(A+1))               04845C00 D
0E9C  4809 CC40 0030 00FC          A2 + B = MIR            % PERFORM ADDITION                   04846C00 D
0E9D  7BC9 0000 0000 00FC          IF AOV THEN SET LC1                                         04847C00 D
0E9E  1E70 0000 0030 C060          CARRY - 1 = CPCR                                            04848C00 D
0E9F  9F1C 0000 0000 0060          CHECKOV - 1 = CPCR                                          04849C00 D
0EA0  49C9 0000 0000 00FC          BMI; SET LC1                                                04850C00 D
0EA1  200C 0000 0000 C060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS            04851C00 D
0EA2  4809 0000 0000 C0F0          BMI                                                         04852C00 D
0EA3  4809 0C41 2000 C0F0          B L = A2                                                    04853C00 D
0EA4  0000 0000 0030 0026          COMP 16 = SAR                                               04854C00 D
0EA5  4809 C000 8030 00FC          A2 R = MIR              % NEW (R(A+1))                      04855C00 D
0EA6  4809 CC40 8800 00F0          B R = B                                                     04856C00 D
0EA7  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR      % WRITE NEW (R(A+1))                04857C00 D
```

```
0EA8  4809 0000 0030 20F0          ASR                      % REF BR1                    04858C00 D
0EA9  4809 0F40 801C 00FC          BMAR R = MAR2                                         04859C00 D
0EAA  000C 00C0 0000 0010          B = SAR                                               04860C00 D
0EAB  4809 0C40 0030 00FC          B = MIR                                               04861C00 D
0EAC  2F50 00C0 0030 C060          EOUTPUT - 1 = CPCR       % WRITE NEW (R(A))           04862C00 D
0EAD  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                     04863C00 D
                                                            %                            04864C00 D
                                                            %                            04865C00 D
                          OP233:                            % TYPE RX ADD DOUBLE, (R(A),R(A+1)) +   04866C00 D
                                                            % (Y,Y+1) INTO R(A),R(A+1)   04867C00 D
                                                            % SET CC, CARRY AND OV BITS  04868C00 D
0EAE  570C 0000 0000 0060          RXMFIELD - 1 = CPCR      % ADDR OF Y INTO B           04869C00 D
0EAF  4809 0C41 0010 00F0          B L = BR2                                             04870C00 D
0EB0  000C 00C0 0030 003C          COMP 8 = SAR                                          04871C00 D
0EB1  50C0 00C0 0000 0060          EMULIN - 1 = CPCR        % (Y) INTO B                04872C00 D
0EB2  4809 0C41 2000 00F0          B L = A2                 % (Y) INTO MS WORD OF A2     04873C00 D
0EB3  0C00 00C0 0030 0020          COMP 16 = SAR                                         04874C00 D
0EB4  4809 0F47 0010 00F0          BMAR + 1 L = BR2                                      04875C00 D
0EB5  0000 00C0 0000 C030          COMP 8 = SAR                                          04876C00 D
0EB6  60E0 00C0 0000 0060          EMULIN - 1 = CPCR        % (Y+1) INTO B              04877C00 D
0EB7  4809 CC5C 0030 00F0          A2 OR B = MIR            % MIR = (Y)/(Y+1)            04878C00 D
0EB8  4809 E000 8B30 00F0          A3 R = B                                              04879C00 D
0EB9  80FC 0000 0000 0000          4 = SAR; 15 = LIT                                     04880C00 D
0EBA  4809 2C5C 0B00 00F0          LIT AND B = B                                         04881C00 D
0EBB  51C0 00C0 0000 C060          REGSTACK - 1 = CPCR                                   04882C00 D
0EBC  4809 0F41 0030 C0F0          BMAR L = BR1             % TEMP STORAGE OF R(A)      04883C00 D
0EBD  0C00 00C0 0030 0030          COMP 8 = SAR                                          04884C00 D
0EBE  2F30 00C0 0030 0060          EINPUT - 1 = CPCR        % (R(A)) INTO B             04885C00 D
0EBF  4809 CC41 2000 00FC          B L = A2                 % (R(A)) INTO MS WORD OF A2  04886C00 D
0EC0  0010 00C0 0030 00A0          COMP 16 = SAR;1 = LIT                                 04887C00 D
0EC1  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2       % R(A+1)                    04888C00 D
0EC2  2F3C 00C0 0000 0060          EINPUT - 1 = CPCR        % (R(A+1)) INTO B            04889C00 D
0EC3  4809 CC5C 2000 00F0          A2 OR B = A2, BMI        % A2 = (R(A))/(R(A+1))       04890C00 D
0EC4  4809 CC40 0030 C0F0          A2 + B = MIR             % PERFORM THE ADDITION       04891C00 D
0EC5  7BC9 00C0 0000 00F0          IF AOV THEN SET LC1                                   04892C00 D
0EC6  1E70 00C0 0000 C060          CARRY - 1 = CPCR                                      04893C00 D
0EC7  4F10 00C0 0030 006C          CHECKOV - 1 = CPCR                                    04894C00 D
0EC8  49C9 00C0 0000 C0F0          BMI; SET LC1                                          04895C00 D
0EC9  200C 00C0 0030 006C          SETCCA - 1 = CPCR        % SET THE CONDITION BITS    04896C00 D
0ECA  4809 0000 0000 00F0          BMI                                                   04897C00 D
0ECB  4809 CC41 2000 00F0          B L = A2                                              04898C00 D
0ECC  000C 00C0 0000 0020          COMP 16 = SAR                                         04899C00 D
0ECD  4809 CC40 8B30 00F0          B R = B                                               04900C00 D
0ECE  4809 C0C0 8C30 C0F0          A2 R = MIR               % (R(A+1))                  04901C00 D
0ECF  2F5C 00C0 0000 0060          EOUTPUT - 1 = CPCR       % WRITE NEW (R(A+1))        04902C00 D
0ED0  4809 00C0 0030 20FC          ASR                      % REF BR1                   04903C00 D
0ED1  4809 0F40 801C 00F0          BMAR R = MAR2                                         04904C00 D
0ED2  0C0C 00C0 0000 0010          B = SAR                                               04905C00 D
0ED3  4809 0C40 0030 00F0          B = MIR                  % (R(A))                     04906C00 D
0ED4  2F5C 00C0 0000 C060          EOUTPUT - 1 = CPCR                                    04907C00 D
0ED5  66E0 0000 0030 0040          OPCODE - 1 = MPCR                                     04908C00 D
                                                            %                            04909C00 D
                                                            %                            04910C00 D
0ED6  5F90 00C0 0000 0060 OPCODE24:  XFCODE - 1 = CPCR      % "F" INTO A2                04911C00 D
0ED7  4809 C640 0030 00FC          A2 + AMPCR = AMPCR                                    04912C00 D
0ED8  17B0 0000 0000 C0C0          OP24F - 1 = AMPCR                                     04913C00 D
0ED9  4809 00C0 0030 0CF0          STEP                                                  04914C00 D
0EDA  4824 00C0 0030 00F0          EXEC                                                  04915C00 D
                                                            %                            04916C00 D
0EDB  17C0 00C0 0000 0040 OP24F:   OP240 - 1 = MPCR                                      04917C00 D
```

```
0EDC  1700 0000 0000 0040          OP241 - 1 = MPCR                                  04918C00 D
0EDD  17EC 0000 0000 C040          OP242 - 1 = MPCR                                  04919C00 D
0EDE  17FC 0000 000C C040          OP243 - 1 = MPCR                                  04920C00 D
                                                          %                          04921C00 D
                          OP240:                          % TYPE RR COMPARE, (R(A)):(R(M))  04922C0C D
                                                          % SET CC, CARRY AND OV BITS  04923C00 D
0EDF  238C 00C0 0C0C C060          CONTENTSRM - 1 = CPCR   % (R(M)) INTO B            04924CC0 D
0EE0  4809 CC40 0000 00F0          B = MIR                 % TEMP STORAGE OF (R(M))    04925C00 D
0EE1  4809 E000 0B0C 00F0          A3 = B                                             04926C00 D
0EE2  22BC 00C0 0C0C C060          CONTENTSRA - 1 = CPCR   % (R(A)) INTO B            04927C00 D
0EE3  4809 CC41 2D0C C0FC          B L = A2, BMI                                      04928C00 D
0EE4  0000 0000 0030 C020          COMP 16 = SAR                                      04929C00 D
0EE5  4809 0C41 0B0C C0F0          B L = B, MIR            % (R(M)) INTO MS WORD OF B, MIR  0493C00C D
0EE6  2000 0000 0000 C06C          SETCC - 1 = CPCR        % SET THE CONDITION BITS   04931C00 D
0EE7  4809 0000 C00C C0F0          BMI                                                04932C00 D
0EE8  4849 CC5E 0C50 C0F0          A2 - B = MIR; SET LC2                              04933C00 D
0EE9  7BC9 C000 000C 00F0          IF AOV THEN SET LC1                                04934C00 D
0EEA  1E7C 00C0 0030 0060          CARRY - 1 = CPCR                                   04935C00 D
0EEB  9F1C 00C0 0000 0060          CHECKOV - 1 = CPCR                                 04936C00 D
0EEC  66EC 00C0 0000 0040          OPCODE - 1 = MPCR                                  04937C06 C
                                                          %                          04938C0C D
                                                          %                          04939C00 D
                          OP241:                          % RI TYPE 2 COMPARE, (R(A)):(Y*)  04940C00 D
                                                          % SET CC, CARRY AND OV BITS  04941C0C D
0EED  22BC 00C0 0030 0060          CONTENTSRA - 1 = CPCR   % (R(A)) INTO B            04942C00 C
0EEE  4809 CC40 0030 C0F0          B = MIR                 % TEMP STORAGE OF (R(A))    04943C00 C
0EEF  238C 00C0 0C0C 0060          CONTENTSRM - 1 = CPCR   % (R(M)) INTO B            04944C00 D
0EF0  4809 CC41 0D10 00F0          B L = BR2, BMI                                     04945C0C D
0EF1  000C 0000 0C30 C030          COMP 8 = SAR                                       04946C00 D
0EF2  4809 CC41 2000 00FC          P L = A2                % (R(A)) INTO A2            04947C00 C
0EF3  0000 00C0 0000 C020          COMP 16 = SAR                                      04948C00 D
0EF4  6CEC 00C0 000C 0060          EMULIN - 1 = CPCR       % (Y*) INTO B              04949C00 D
0EF5  4809 0C41 0B30 00F0          P L = B, MIR            % (Y*) INTO MS WORD OF B , MIR  04950C00 D
0EF6  0000 C0C0 0030 0020          COMP 16 = SAR                                      04951C00 D
0EF7  2000 00C0 0C0C 0060          SETCC - 1 = CPCR        % SET THE CONDITION BITS   04952C00 D
0EF8  4809 00C0 000C C0FC          BMI                     % (Y*) BACK INTO B         04953C00 D
0EF9  4849 CC5E 0030 00FC          A2 - B = MIR; SET LC2                              04954C00 D
0EFA  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                                04955C0C D
0EFB  1E7C 00C0 000C 0060          CARRY - 1 = CPCR                                   04956C00 D
0EFC  9F1C 00C0 0000 0060          CHECKOV - 1 = CPCR                                 04957C00 D
0EFD  66ED 00C0 0000 C04C          OPCODE - 1 = MPCR                                  04958C00 D
                                                          %                          04959C00 D
                                                          %                          04960C00 D
                          OP242:                          % TYPE RK COMPARE, (R(A)):Y 04961C00 D
                                                          % SET CC, CARRY AND OV BITS  04962C00 D
0EFE  4809 2C55 0B30 C0F0          LIT AND B = B                                      04963C0C D
0EFF  00FC 00C0 0C30 00E0          15 = LIT                                           04964C00 D
0F00  4809 CC52 0000 00FC          Q EQL B                                            04965C00 D
0F01  6819 00C0 0030 C0F0          IF TRUE THEN SKIP                                  04966C00 D
0F02  238C 00C0 0000 0060          CONTENTSRM - 1 = CPCR   % (R(M)) INTO B            04967C00 D
0F03  4809 0C41 0B0C C0F0          B L = B                                            04968C0C D
0F04  0000 0000 0C30 C020          COMP 16 = SAR                                      04969C00 D
0F05  4809 EC5C 100C C0F0          A3 OR B = A3            % A3 = (R(M))/INSTRUCTION  04970C00 D
0F06  533C 00C0 0030 0060          IFETCH - 1 = CPCR       % *Y* INTO B               04971C00 D
0F07  4809 E0C0 A030 C0F0          A3 R = A2               % (R(M)) INTO A2           04972C0C D
0F08  0000 0000 0C0C 0020          16 = SAR                                           04973C00 D
0F09  4809 CC41 0C50 00F0          A2 + B L = MIR          % Y INTO MS WORD OF MIR    04974C00 D
0F0A  4809 E0C0 0B3C 00F0          A3 = B                                             04975C00 D
0F0B  22BC 00C0 0C0C C060          CONTENTSRA - 1 = CPCR   % (R(A)) INTO B            04976C00 D
0F0C  4809 0C41 2D0C 00F0          B L = A2, BMI           % (R(A)) INTO MS WORD OF A2  04977C00 D
```

```
0F0D  0000 0000 0000 0020          COMP 16 = SAR                                  04978000 D
0F0E  4809 0040 0090 00F0          B = MIR                                        04979000 D
0F0F  2000 0000 0000 0060          SETCC - 1 = CPCR        % SET THE CONDITION BITS 04980000 D
0F10  4809 0000 0000 00F0          BMI                                            04981000 D
0F11  4849 0050 0080 00F0          A2 - B = MIR; SET LC2                          04982000 D
0F12  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                            04983000 D
0F13  1E70 0000 0000 0060          CARRY - 1 = CPCR                               04984000 D
0F14  4F10 0000 0000 0060          CHECKOV - 1 = CPCR                             04985000 D
0F15  66E0 0000 0000 0040          OPCODE - 1 = MPCR                              04986000 D
                                                          %                       04987000 D
                                                          %                       04988000 D
                                   OP243:                 % RX TYPE COMPARE, (R(A)):Y 04989000 D
                                                          % SET CC, CARRY, AND OV BITS 04990000 D
0F16  5700 0000 0000 0060          RXMFIELD - 1 = CPCR     % Y INTO B              04991000 D
0F17  4809 0041 0010 00F0          B L = BR2                                      04992000 D
0F18  0000 0000 0000 0030          COMP 8 = SAR                                   04993000 D
0F19  60E0 0000 0000 0060          EMULIN - 1 = CPCR       % (Y) INTO B           04994000 D
0F1A  4809 0041 0030 00F0          B L = MIR                                      04995000 D
0F1B  0000 0000 0000 0020          COMP 16 = SAR                                  04996000 D
0F1C  4809 0000 0800 00F0          A3 = B                                         04997000 D
0F1D  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR   % (R(A)) INTO B        04998000 D
0F1E  4809 0041 2000 00F0          B L = A2, BMI                                  04999000 D
0F1F  0000 0000 0000 0020          COMP 16 = SAR                                  05000000 D
0F20  4809 0040 0090 00F0          B = MIR                                        05001000 D
0F21  2000 0000 0000 0060          SETCC - 1 = CPCR                               05002000 D
0F22  4809 0000 0000 00F0          BMI                                            05003000 D
0F23  4849 0050 0080 00F0          A2 - B = MIR; SET LC2                          05004000 D
0F24  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                            05005000 D
0F25  1E70 0000 0000 0060          CARRY - 1 = CPCR                               05006000 D
0F26  4F10 0000 0000 0060          CHECKOV - 1 = CPCR                             05007000 D
0F27  66E0 0000 0000 0040          OPCODE - 1 = MPCR                              05008000 D
                                                          %                       05009000 D
                                                          %                       05010000 D
0F28  5F90 0000 0000 0060          OPCODE25:  XFCODE - 1 = CPCR  % "F" INTO A2    05011000 D
0F29  4809 0640 0040 00F0          A2 + AMPCR = AMPCR                             05012000 D
0F2A  1800 0000 0000 00C0          OP25F - 1 = AMPCR                              05013000 D
0F2B  4809 0000 0000 00F0          STEP                                           05014000 D
0F2C  4824 0000 0000 00F0          EXEC                                           05015000 D
                                                          %                       05016000 D
0F2D  1810 0000 0000 0040          OP25F:  OP250 - 1 = MPCR                       05017000 D
0F2E  1820 0000 0000 0040          OP251 - 1 = MPCR                               05018000 D
0F2F  3000 0000 0000 0040          FAULT - 1 = MPCR                               05019000 D
0F30  1830 0000 0000 0040          OP253 - 1 = MPCR                               05020000 D
                                                          %                       05021000 D
                                   OP250:                 % TYPE RR COMPARE DOUBLE REGISTER 05022000 D
                                                          % (R(A),R(A+1)) : (R(M),R(M+1)), 05023000 D
                                                          % SET CARRY, CC, AND OV BITS 05024000 D
0F31  4809 0040 8800 00F0          B R = B                                        05025000 D
0F32  80F0 0000 0000 008C          15 = LIT; 4 = SAR                              05026000 D
0F33  4809 2056 0800 00F0          LIT AND B = B           % ISOLATE "A" FIELD    05027000 D
0F34  51C0 6000 0000 0060          REGSTACK - 1 = CPCR                            05028000 D
0F35  2F30 6000 0000 006C          EINPUT - 1 = CPCR       % (R(A)) INTO B        05029000 D
0F36  4809 0041 2000 00F0          B L = A2                                       05030000 D
0F37  0010 0000 0000 00A0          COMP 16 = SAR;1 = LIT                          05031000 D
0F38  4809 2F50 0010 00F0          LIT OR BMAR = MAR2      % R(A+1)               05032000 D
0F39  2F30 0000 0000 0060          EINPUT - 1 = CPCR       % (R(A+1)) INTO B      05033000 D
0F3A  4809 0050 2000 00F0          A2 OR B = A2            % A2 = (R(A))/(R(A+1)) 05034000 D
0F3B  4809 E000 0800 00F0          A3 = B                                         05035000 D
0F3C  4809 2056 0800 00F0          LIT AND B = B           % ISOLATE "M" FIELD     05036000 D
0F3D  00F0 0000 0000 00E0          15 = LIT                                       05037000 D
```

```
0F3E  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                          05038000 D
0F3F  2F30 0000 0030 0060          EINPUT - 1 = CPCR      % (R(M)) INTO B       05039000 D
0F40  4809 0C41 1030 00F0          B L = A3                                     05040000 D
0F41  0010 0000 0000 00A0          COMP 16 = SAR; 1 = LIT                       05041000 D
0F42  4809 2F5C 0C1C 00F0          LIT OR BMAR = MAR2     % ADDR OF R(M+1)      05042000 D
0F43  2F30 0000 0030 0060          EINPUT - 1 = CPCR      % (R(M+1)) INTO B     05043000 D
0F44  4809 EC5C 0B30 00F0          A3 OR B = B, MIR       % B = (R(M))/(R(M+1))  05044000 D
0F45  2000 0000 0000 0060          SETCC - 1 = CPCR                             05045000 D
0F46  4809 0000 0030 00F0          BMI                                          05046000 D
0F47  4849 CC5E 0080 00F0          A2 - B = MIR; SET LC2                        05047000 D
0F48  7809 0000 0000 00F0          IF AOV THEN SET LC1                          05048000 D
0F49  1E70 0000 0000 0060          CARRY - 1 = CPCR                             05049000 D
0F4A  9F10 0000 0000 0060          CHECKOV - 1 = CPCR                           05050000 D
0F4B  66E0 0000 0000 0040          OPCODE - 1 = MPCR                            05051000 D
                                                          %                     05052000 D
                                                          %                     05053000 D
                              OP251:                      % RI TYPE 2 COMPARE DOUBLE 05054000 D
                                                          % (R(A),R(A+1)):(Y*,Y*+1) 05055000 D
                                                          % SET CC, CARRY, AND OV BITS 05056000 D
0F4C  4809 0C40 8D30 00F0          B R = B                                      05057000 D
0F4D  90FC 0000 0030 0080          4 = SAR; 15 = LIT                            05058000 D
0F4E  4809 2C56 0B00 00F0          LIT AND B = B          % ISOLATE "A" FIELD   05059000 D
0F4F  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                          05060000 D
0F50  2F30 0000 0030 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B       05061000 D
0F51  4809 0C41 2000 00F0          B L = A2               % (R(A)) INTO MS WORD OF A2 05062000 D
0F52  0010 0000 0000 00A0          COMP 16 = SAR ; 1 = LIT                      05063000 D
0F53  4809 2F5C 0C1C 00F0          LIT OR BMAR = MAR2     % R(A+1)              05064000 D
0F54  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B     05065000 D
0F55  4809 CC5C 2030 00F0          A2 OR B = A2           % A2 = (R(A))/(RA+1))  05066000 D
0F56  4809 E156 0B30 00F0          A3 AND LIT = B         % ISOLATE "M" FIELD    05067000 D
0F57  00F0 0000 0030 00E0          15 = LIT                                     05068000 D
0F58  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                          05069000 D
0F59  4809 0F41 0010 00F0          BMAR L = BR2                                 05070000 D
0F5A  0000 0000 0030 0030          COMP B = SAR                                 05071000 D
0F5B  50E0 0000 0000 006C          EMULIN - 1 = CPCR      % (Y*) INTO B         05072000 D
0F5C  4809 0C41 1030 00F0          B L = A3               % (Y*) INTO MS WORD OF A3 05073000 D
0F5D  0010 0000 0030 00A0          COMP 16 = SAR ; 1 = LIT                      05074000 D
0F5E  4809 2F53 001C 00F0          LIT OR BMAR L = MAR2   % ADDR OF Y* + 1      05075000 D
0F5F  0000 0000 0030 0030          COMP B = SAR                                 05076000 D
0F60  60E0 0000 0030 0060          EMULIN - 1 = CPCR      % (Y*+1) INTO B       05077000 D
0F61  4809 EC5C 0B30 00F0          A3 OR B = B, MIR       % B = (Y*)/(Y*+1)      05078000 D
0F62  2000 0000 0000 0060          SETCC - 1 = CPCR       % SET THE CONDITION BITS 05079000 D
0F63  4809 0000 0030 00F0          BMI                                          05080000 D
0F64  4849 CC5E 0080 00F0          A2 - B = MIR; SET LC2                        05081000 D
0F65  7809 0000 0000 00F0          IF AOV THEN SET LC1                          05082000 D
0F66  1E70 0000 0000 0060          CARRY - 1 = CPCR                             05083000 D
0F67  9F10 0000 0000 0060          CHECKOV - 1 = CPCR                           05084000 D
0F68  66E0 0000 0000 0040          OPCODE - 1 = MPCR                            05085000 D
                                                          %                     05086000 D
                                                          %                     05087000 D
                              OP253:                      % TYPE RX DOUBLE COMPARE 05088000 D
                                                          %(R(A),R(A+1)):(Y,Y+1) 05089000 D
                                                          % SET CC, CARRY AND OV BITS 05090000 D
0F69  5700 0000 0000 0060          RXMFIELD - 1 = CPCR     % Y INTO E            05091000 D
0F6A  4809 0C41 0010 00F0          B L = BR2                                    05092000 D
0F6B  0000 0000 0030 0030          COMP B = SAR                                 05093000 D
0F6C  60E0 0000 0030 0060          EMULIN - 1 = CPCR      % (Y) INTO B          05094000 D
0F6D  4809 0C41 2000 00F0          B L = A2               % (Y) INTO MS WORD OF A2 05095000 D
0F6E  0010 0000 0030 00A0          COMP 16 = SAR ; 1 = LIT                      05096000 D
0F6F  4809 2F53 0010 00F0          LIT OR BMAR L = BR2    % ADDR OF Y+1         05097000 D
```

```
0F70  0000 0000 0000 0030          COMP B = SAR                          % (Y+1) INTO B                    05098000 D
0F71  60E0 0000 0000 0060          EMULIN - 1 = CPCR                     % MIR = (Y)/(Y+1)                 05099000 D
0F72  4809 CC5C 0C80 00F0          A2 OR B = MIR                                                          05100000 D
0F73  4809 E0C0 9000 C0F0          A3 R = A3                                                              05101000 D
0F74  50F0 0000 0000 C08C          4 = SAR; 15 = LIT                                                      05102000 D
0F75  4809 E156 0B0C 00F0          A3 AND LIT = B                        % ISOLATE "A"                     05103000 D
0F76  51C0 C000 C030 C060          REGSTACK - 1 = CPCR                                                    05104000 D
0F77  2F30 00C0 C00C 0060          EINPUT - 1 = CPCR                     % (R(A)) INTO B                  05105000 D
0F78  4809 0C41 2000 00F0          B L = A2                              % (R(A)) INTO MS WORD OF A2       05106000 D
0F79  001C 00C0 0000 C0AC          COMP 16 = SAR ; 1 = LIT                                                05107000 D
0F7A  4809 2F5C C01C C0FC          LIT OR BMAR = MAR2                    % R(A+1)                          05108000 D
0F7B  2F30 0C03 0000 C06C          EINPUT - 1 = CPCR                     % (R(A+1)) INTO B                 05109000 D
0F7C  4809 CC5C 2D9C 00F0          A2 OR B = A2, BMI                     % A2 = (R(A))/(R(A+1))            05110000 D
0F7D  2000 0000 0000 C06C          SETCC - 1 = CPCR                      % SET THE CONDITION BITS          05111000 D
0F7E  4809 C0C0 C030 00F0          BMI                                                                    05112000 D
0F7F  4849 CC5E 0090 00FC          A2 - B = MIR; SET LC2                                                  05113000 D
0F80  7BC9 00C0 0000 00F0          IF ADV THEN SET LC1                                                    05114000 D
0F81  1E7C 0C00 C00C 0060          CARRY - 1 = CPCR                                                       05115000 D
0F82  4F1C 00C0 0C9C C060          CHECKOV - 1 = CPCR                                                     05116000 D
0F83  66EC C0C0 000C 0040          OPCODE - 1 = MPCR                                                      05117000 D
                                                                         %                                 05118000 D
                                                                         %                                 05119000 D
0F84  5F9C 0C00 0030 0060  OPCODE26:  XFCODE - 1 = CPCR                  % "F" INTO A2                     05120000 D
0F85  4809 C640 C030 00F0          A2 + AMPCR = AMPCR                                                     05121000 D
0F86  184C 00C0 C00C 00C0          OP26F - 1 = AMPCR                                                      05122000 D
0F87  4809 00C0 0030 C0F0          STCP                                                                   05123000 D
0F88  4824 0000 0030 C0F0          EXEC                                                                   05124000 D
                                                                         %                                 05125000 D
0F89  1850 00C0 0000 0040  OP26F:  OP260 - 1 = MPCR                                                       05126000 D
0F8A  1860 C0C0 0C00 0040          OP261 - 1 = MPCR                                                       05127000 D
0F8B  187C 00C0 0000 0040          OP262 - 1 = MPCR                                                       05128000 D
0F8C  1880 0000 0000 C040          OP263 - 1 = MPCR                                                       05129000 D
                                                                         %                                 05130000 D
                                                                         %                                 05131000 D
                           OP260:                                        % RR TYPE MULTIPLY (REGISTER)      05132000 D
                                                                         % (R(A+1)) X (R(M)) = R(A),R(A+1), SET CC05133000 D
0F8D  4809 0C40 0B30 00F0          B R = B                                                                05134000 D
0F8E  80FC 00C0 0000 C08C          4 = SAR; 15 = LIT                                                      05135000 D
0F8F  4809 2C56 0B00 00FC          LIT AND B = B                         % ISOLATE "A"                     05136000 D
0F90  51C0 C0C0 0030 006C          REGSTACK - 1 = CPCR                                                    05137000 D
0F91  4809 0F41 C030 00F0          BMAR L = BR1                          % TEMP STORAGE                    05138000 D
0F92  000C 00C0 0C00 0C30          COMP B = SAR                                                           05139000 D
0F93  4809 0F45 0B3C C0FC          BMAR + 1 = B                          % R(A+1)                          05140000 D
0F94  51C0 00C0 0C00 006C          REGSTACK - 1 = CPCR                   % R(A+1) INTO MAR2                05141000 D
0F95  2F3C 0003 0000 C06C          EINPUT - 1 = CPCR                     % (R(A+1)) INTO B                 05142000 D
0F96  4809 0C40 2C30 00F0          B = A2                                                                 05143000 D
0F97  4809 E155 0B0C 00FC          A3 AND LIT = B                        % ISOLATE "M"                     05144000 D
0F98  00FC 00C0 C03C 00L0          15 = LIT                                                               05145000 D
0F99  51C0 0003 CC00 C060          REGSTACK - 1 = CPCR                   % R(M) INTO MAR2                  05146000 D
0F9A  2F30 00C0 0030 0060          EINPUT - 1 = CPCR                     % (R(M)) INTO E                   05147000 D
0F9B  4C6C C0C0 0C0C C060          MULT - 1 = CPCR                       % MULT (R(A+1)) X (R(M))          05148000 D
0F9C  4809 E0D0 0B3C C0FC          A3 = B                                % PRODUCT IN A3                   05149000 D
0F9D  200C 00C0 0C3C C06C          SETCCA - 1 = CPCR                     % SET THE CONDITION BITS          05150000 D
0F9E  4809 E0D0 8C30 00FC          A3 R = MIR                            % (R(A))                          05151000 D
0F9F  000C C0C3 0030 0020          16 = SAR                                                               05152000 D
0FA0  4809 00C0 0030 20F0          ASR                                   % REF ER1                         05153000 D
0FA1  4809 0F40 801C C0FC          BMAR R = MAR2                         % R(A)                            05154000 D
0FA2  0C0C 00C0 0030 C010          6 = SAR                                 \                              05155000 D
0FA3  2F5C 0000 0030 0060          EOUTPUT - 1 = CPCR                                                     05156000 D
0FA4  4809 0F46 0B3C C0FC          BMAR + 1 = B                          % R(A+1)                          05157000 D
```

```
0FA5  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2                  05156000 D
0FA6  4809 E001 1000 00F0          A3 L = A3                                                  05159000 D
0FA7  0000 00C0 0C30 0020          COMP 16 = SAR                                              05160000 D
0FA8  4809 E0C0 8C30 C0F0          A3 R = MIR                                                 05161000 D
0FA9  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                         05162000 D
0FAA  66E0 0000 0000 004C          OPCODE - 1 = MPCR                                          05163000 D
                                                          %                                   05164000 D
                                                          %                                   05165000 D
                              OP261:                      % RI TYPE 2 MULTIPLY (INDIRECT)      05166000 D
                                                          % (R(A+1)) X (Y*) = R(A),R(A+1), SET CC 05167000 D
0FAB  4809 0C40 8800 00F0          B R = B                                                    05168000 D
0FAC  80FC C000 0000 0080          4 = SAR, 15 = LIT                                          05169000 D
0FAD  4809 2C55 0800 00F0          LIT AND B = B          % ISOLATE "A"                       05170000 D
0FAE  51C0 0000 0000 006C          REGSTACK - 1 = CPCR                                        05171000 D
0FAF  4809 0F41 0020 C0F0          BMAR L = BR1           % R(A)                              05172000 D
0FB0  0000 0000 0000 003C          COMP 8 = SAR                                               05173000 D
0FB1  4809 0F46 0E00 C0F0          BMAR + 1 = B           % R(A+1)                            05174000 D
0FB2  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2                  05175000 D
0FB3  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B                   05176000 D
0FB4  4809 0C40 2000 00F0          B = A2                 % (R(A+1)) INTO A2                  05177000 D
0FB5  4809 E156 0800 C0F0          A3 AND LIT = B         % ISOLATE "M"                       05178000 D
0FB6  00F0 00C0 0000 C0E0          15 = LIT                                                   05179000 D
0FB7  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(M) INTO MAR2                    05180000 D
0FB8  2F30 00C0 0000 C060          EINPUT - 1 = CPCR      % Y* INTO B                         05181000 D
0FB9  4809 0C41 0010 C0F0          B L = BR2                                                  05182000 D
0FBA  0000 0000 0000 0030          COMP 8 = SAR                                               05183000 D
0FBB  60E0 0000 0000 C060          EMULIN - 1 = CPCR      % (Y*) INTO B                      05184000 D
0FBC  4C6C 0000 0000 C060          MULT - 1 = CPCR        % MULT (R(A+1)) X (Y*)             05185000 D
0FBD  4809 E000 0800 C0FC          A3 = B                 % PRODUCT IN A3                     05186000 D
0FBE  20D0 00C0 0C30 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS            05187000 D
0FBF  4809 E0C0 8030 C0F0          A3 R = MIR             % (R(A))                            05188000 D
0FC0  0000 0000 0000 C020          16 = SAR                                                   05189000 D
0FC1  4809 0000 0000 20FC          ASR                    % REF ER1                           05190000 D
0FC2  4809 0F43 801C 00F0          BMAR R = MAR2          % R(A)                              05191000 D
0FC3  000C 00C0 0C30 0010          B = SAR                                                    05192000 D
0FC4  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                         05193000 D
0FC5  4809 0F46 0800 00F0          BMAR + 1 = B           % R(A+1)                            05194000 D
0FC6  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2                  05195000 D
0FC7  4809 E001 1000 00FC          A3 L = A3                                                  05196000 D
0FC8  0000 00C0 0000 0020          COMP 16 = SAR                                              05197000 D
0FC9  4809 E0C0 8030 C0F0          A3 R = MIR                                                 05198000 D
0FCA  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                         05199000 D
0FCB  66EC C0C0 0C30 C040          OPCODE - 1 = MPCR                                          05200000 D
                                                          %                                   05201000 D
                                                          %                                   05202000 D
                              OP262:                      % TYPE RK MULTIPLY (CONSTANT)        05203000 D
                                                          % (R(A+1)) X Y = R(A),R(A+1), SET CC 05204000 D
0FCC  4809 2C56 0800 00F0          LIT AND B = B          % ISOLATE "M"                       05205000 D
0FCD  00F0 0000 0000 00EC          15 = LIT                                                   05206000 D
0FCE  4809 0C52 0030 00F0          0 EQL B                % CHECK FOR M = 0                   05207000 D
0FCF  5819 C000 0C30 C0F0          IF TRUE THEN SKIP                                          05208000 D
0FD0  238C 0000 0000 C060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                    05209000 D
0FD1  4809 0C41 0D10 00F0          B L = B                                                    05210000 D
0FD2  0000 0000 0030 C020          COMP 16 = SAR                                              05211000 D
0FD3  4809 EC5C 1000 C0F0          A3 OR B = A3           % A3 = (R(M))/INSTRUCTION           05212000 D
0FD4  6330 0000 0C30 C060          IFETCH - 1 = CPCR      % "Y" INTO B                        05213000 D
0FD5  4809 E000 A000 00F0          A3 R = A2              % (R(M)) INTO A2                    05214000 D
0FD6  0000 0000 0000 CC20          16 = SAR                                                   05215000 D
0FD7  4809 0C40 2C30 C0FC          A2 + B = A2            % Y INTO B                          05216000 D
0FD8  4809 E0C0 8B00 00F0          A3 R = B                                                   05217000 D
```

```
0FD9  3CFC 0000 0000 0080          4 = SAR; 15 = LIT
0FDA  4809 2056 0800 00F0          LIT AND B = B              % ISOLATE "A"                  05219C00 D
0FDB  51CC 0000 0000 0060          REGSTACK - 1 = CPCR                                       05220C00 D
0FDC  4809 0F41 0020 C0F0          BMAR L = BR1               % R(A)                         05221C00 D
0FDD  0000 0000 0000 C030          COMP 8 = SAR                                              05222C00 D
0FDE  4809 0F46 0800 00F0          BMAR + 1 = B               % R(A+1)                       05223C00 D
0FDF  51C0 0000 0000 C060          REGSTACK - 1 = CPCR        % R(A+1) INTO MAR2             05224C00 D
0FE0  2F3C 0000 0000 006C          EINPUT - 1 = CPCR          % (R(A+1)) INTO B              05225C00 D
0FE1  4C60 0000 0000 0060          MULT - 1 = CPCR            % MULTIPLY (R(A+1)) X  Y       05226C00 D
0FE2  4809 E000 0800 00F0          A3 = B                     % PRODUCT IN A3                 05227C00 D
0FE3  2000 0000 0000 006C          SETCCA - 1 = CPCR          % SET THE CONDITION BITS        05228C00 D
0FE4  4809 E000 8C30 00F0          A3 R = MIR                 % (R(A))                        05229C00 D
0FE5  0000 0000 0030 0020          16 = SAR                                                  05230C00 D
0FE6  4809 0000 0000 20F0          ASR                        % REF BR1                      05231C00 D
0FE7  4809 0F43 801C C0F0          BMAR R = MAR2              % R(A)                         05232C00 D
0FE8  0000 0000 0000 0010          8 = SAR                                                   05233C00 D
0FE9  2F5C 0000 0000 C060          EOUTPUT - 1 = CPCR                                        05234C00 D
0FEA  4809 0F45 0800 C0F0          BMAR + 1 = B               % R(A+1)                       05235C00 D
0FEB  51C0 0000 0000 0060          REGSTACK - 1 = CPCR        % R(A+1) INTO MAR2             05236C00 D
0FEC  4809 E001 1000 C0F0          A3 L = A3                                                 05237C00 D
0FED  0000 0000 0000 0020          COMP 16 = SAR                                             05238C00 D
0FEE  4809 E000 8030 00F0          A3 R = MIR                                                05239C00 D
0FEF  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                        05240C00 D
0FF0  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                         05241C00 D
                                                              %                              05242C00 D
                                                              %                              05243C00 D
                        OP263:                                % RX TYPE MULTIPLY, (R(A+1)) X (Y) =  05244C00 D
                                                              % R(A),R(A+1), AND SET CC       05245C00 D
0FF1  5700 0000 0000 C060          RXMFIELD - 1 = CPCR        % Y INTO B                     05246C00 D
0FF2  4809 0C41 0010 C0F0          B L = BR2                                                 05247C00 D
0FF3  0000 0000 0000 0030          COMP 8 = SAR                                              05248C00 D
0FF4  60E0 0000 0000 0060          EMULIN - 1 = CPCR          % (Y) INTO B                   05249C00 D
0FF5  4809 0C40 0030 C0F0          B = MIR                                                   05250C00 D
0FF6  4809 E000 8800 00F0          A3 R = B                                                  05251C00 D
0FF7  80FC 0000 0000 0080          4 = SAR; 15 = LIT                                         05252C00 D
0FF8  4809 2056 0800 00F0          LIT AND B = B              % ISOLATE "A"                  05253C00 D
0FF9  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                                       05254C00 D
0FFA  4809 0F41 0020 C0F0          BMAR L = BR1               % SAVE R(A)                    05255C00 D
0FFB  0000 0000 0000 0030          COMP 8 = SAR                                              05256C00 D
0FFC  4809 0F46 0800 00F0          BMAR + 1 = B               % R(A+1)                       05257C00 D
0FFD  51C0 0000 0000 0060          REGSTACK - 1 = CPCR        % R(A+1) INTO MAR2             05258C00 D
0FFE  2F3C 0000 0000 006C          EINPUT - 1 = CPCR          % (R(A+1)) INTO B              05259C00 D
0FFF  4809 0C40 2000 00F0          B = A2, BMI                                               05260C00 D
1000  4C6C 0000 0000 C060          MULT - 1 = CPCR            % MULTIPLY (R(A+1)) X (Y)      05261C00 D
1001  4809 E000 C800 00F0          A3 = B                     % PRODUCT IN A3                 05262C00 D
1002  2000 0000 0000 0060          SETCCA - 1 = CPCR          % SET THE CONDITION BITS        05263C00 D
1003  4809 E000 8030 C0F0          A3 R = MIR                 % (R(A))                        05264C00 D
1004  0000 0000 0000 0020          16 = SAR                                                  05265C00 D
1005  4809 0000 0000 20F0          ASR                        % REF BR1                      05266C00 D
1006  4809 0F43 801C C0F0          BMAR R = MAR2              % R(A)                         05267C00 D
1007  0000 0000 0000 0010          8 = SAR                                                   05268C00 D
1008  2F50 0000 0000 006C          EOUTPUT - 1 = CPCR                                        05269C00 D
1009  4809 0F46 0800 C0F0          BMAR + 1 = B               % R(A+1)                       05270C00 D
100A  51C0 0000 0000 C060          REGSTACK - 1 = CPCR        % R(A+1) INTO MAR2             05271C00 D
100B  4809 E001 1000 00F0          A3 L = A3                                                 05272C00 D
100C  4809 E000 8030 00F0          A3 R = MIR                                                05273C00 D
100D  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                        05274C00 D
100E  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                         05275C00 D
                                                              %                              05276C00 D
                                                              %                              05277C00 D
```

```
100F  5F90 0000 0000 C060   OPCODE27:  XFCODE - 1 = CPCR   % "F" INTO A2                    05278000 D
1010  4809 0640 0000 00F0              A2 + AMPCR = AMPCR                                   05279000 D
1011  1890 0000 0000 0000              OP27F - 1 = AMPCR                                    05280000 D
1012  4809 0000 0000 00F0              STEP                                                 05281000 D
1013  4824 0000 0000 00F0              EXEC                                                 05282000 D
                                                          %                                  05283000 D
1014  18A0 0000 0000 0040   OP27F:     OP270 - 1 = MPCR                                     05284000 D
1015  18B0 0000 0000 0040              OP271 - 1 = MPCR                                     05285000 D
1016  18C0 0000 0000 0040              OP272 - 1 = MPCR                                     05286000 D
1017  18D0 0000 0000 0040              OP273 - 1 = MPCR                                     05287000 D
                                                          %                                  05288000 D
                                                          %                                  05289000 D
                            OP270:                        % TYPE RR DIVIDE (REGISTER)         05290000 D
                                                          % R(A),R(A+1)/(R(M)) = R(A+1), REMAINDER 05291000 D
                                                          % INTO R(A), SET CC AND OV          05292000 D
1018  4809 0040 8B30 00F0              B R = B                                              05293000 D
1019  80F0 0000 0000 0080              4 = SAR; 15 = LIT                                    05294000 D
101A  4809 2056 0B30 00F0              LIT AND B = B       % ISOLATE "A"                    05295000 D
101B  51C0 0000 0000 0060              REGSTACK - 1 = CPCR                                  05296000 D
101C  4809 0F41 0020 00F0              BMAR L = BR1        % TEMP STORAGE                   05297000 D
101D  0000 0000 0000 0030              COMP 8 = SAR                                         05298000 D
101E  2F30 0000 0000 0060              EINPUT - 1 = CPCR   % (R(A)) INTO B                  05299000 D
101F  4809 0041 2000 00F0              B L = A2                                             05300000 D
1020  0000 0000 0000 0020              COMP 16 = SAR                                        05301000 D
1021  4809 0F46 0B30 00F0              BMAR + 1 = B        % R(A+1)                         05302000 D
1022  51C0 0000 0000 0060              REGSTACK - 1 = CPCR % R(A+1) INTO MAR2               05303000 D
1023  2F30 0000 0000 0060              EINPUT - 1 = CPCR   % (R(A+1)) INTO B                05304000 D
1024  4809 0C50 2000 00F0              A2 OR B = A2        % A2 = (R(A))/(R(A+1))            05305000 D
1025  4809 E156 0B30 00F0              A3 AND LIT = B      % ISOLATE "M"                     05306000 D
1026  00F0 0000 0000 00EC              15 = LIT                                             05307000 D
1027  51C0 0000 0000 0060              REGSTACK - 1 = CPCR                                  05308000 D
1028  2F30 0000 0000 006C              EINPUT - 1 = CPCR   % (R(M)) INTO E                  05309000 D
1029  49C9 0041 0B30 00F0              B L = B; SET LC1    % LEFT JUSTIFY (R(M)), SET 32 BIT FLAG 05310000 D
102A  0000 0000 0000 0020              COMP 16 = SAR                                        05311000 D
102B  240C 0000 0000 0060              DIV - 1 = CPCR      % (R(A),R(A+1))/(R(M))            05312000 D
102C  2DC0 0000 0000 00F0              IF LC1 THEN SET LC1; STEP ELSE SKIP  % CHECK FOR OVERFLOW 05313000 D
102D  50A0 0000 0000 0060              SETOVBIT - 1 = CPCR % SET THE OV BIT                 05314000 D
102E  2819 00C0 0000 00F0              IF LC1 THEN SKIP                                     05315000 D
102F  5050 0000 0000 0060              CLEAROV - 1 = CPCR  % CLEAR THE OV BIT               05316000 D
1030  4809 E0C0 0B30 00F0              A3 = B              % QUOTIENT INTO B                05317000 D
1031  4809 C0C0 0030 00F0              A2 = MIR            % REMAINDER INTO MIR             05318000 D
1032  2000 0000 0000 0060              SETCCA - 1 = CPCR   % SET THE CONDITION BITS         05319000 D
1033  4809 0000 0000 20F0              ASR                 % REFERENCE BR1                   05320000 D
1034  4809 0F40 8010 00F0              BMAR R = MAR2       % R(A)                           05321000 D
1035  0000 0000 0000 0010              8 = SAR                                              05322000 D
1036  2F50 0000 0000 0060              EOUTPUT - 1 = CPCR  % REMAINDER IN R(A)              05323000 D
1037  4809 0000 0000 24F0              ASE                 % REFERENCE MAR2                  05324000 D
1038  4809 0F46 0B00 00F0              BMAR + 1 = B        % CREATE R(A+1)                   05325000 D
1039  51C0 0000 0000 0060              REGSTACK - 1 = CPCR % R(A+1) INTO MAR2               05326000 D
103A  4809 E0C0 0030 00F0              A3 = MIR            % QUOTIENT                       05327000 D
103B  2F50 0000 0000 0060              EOUTPUT - 1 = CPCR                                   05328000 D
103C  66E0 0000 0000 0040              OPCODE - 1 = MPCR                                    05329000 D
                                                          %                                  05330000 D
                                                          %                                  05331000 D
                            OP271:                        % RI TYPE 2 DIVIDE (INDIRECT)       05332000 D
                                                          % (R(A),R(A+1))/(Y*) = R(A+1), REMAINDER 05333000 D
                                                          % INTO R(A), SET CC AND OV          05334000 D
103D  4809 0040 8B30 00F0              B R = B                                              05335000 D
103E  80F0 0000 0000 0080              4 = SAR; 15 = LIT                                    05336000 D
103F  4809 2056 0B30 00F0              LIT AND B = B       % ISOLATE "A"                    05337000 D
```

```
1C40  31C0 00C0 C030 0060   REGSTACK - 1 = CPCR                                  05338000 D
1C41  4809 0F41 0020 00F0   BMAR L = BR1          % TEMP STORAGE                05339000 D
1C42  0C00 00C0 0000 C030   COMP 8 = SAR                                        05340000 D
1C43  2F30 00C0 0030 0060   EINPUT - 1 = CPCR     % (R(A)) INTO B               05341000 D
1C44  4809 0C41 2000 00F0   B L = A2                                            05342000 D
1C45  00C0 0000 0030 C020   COMP 16 = SAR                                       05343000 D
1C46  4809 0F46 0B30 00F0   BMAR + 1 = B          % R(A+1)                      05344000 D
1C47  31C0 00C0 0030 0060   REGSTACK - 1 = CPCR   % P(A+1) INTO MAR2            05345000 D
1C48  2F30 00C0 0030 C060   EINPUT - 1 = CPCR     % (R(A+1)) INTO B             05346000 D
1C49  4809 CC50 2C00 00F0   A2 OR B = A2          % A2 =(R(A))/(R(A+1))         05347000 D
1C4A  4809 E156 0B00 00F0   A3 AND LIT = B        % ISOLATE "M"                 05348000 D
1C4B  00F0 00C0 0000 00E0   15 = LIT                                            05349000 D
1C4C  31C0 00C0 0030 0060   REGSTACK - 1 = CPCR                                 05350000 D
1C4D  2F30 00C0 0030 C060   EINPUT - 1 = CPCR     % Y" INTO B                   05351000 D
1C4E  4809 0C41 001C 00F0   B L = BR2                                           05352000 D
1C4F  0C00 00C0 0000 C030   COMP 8 = SAR                                        05353000 D
1C50  50E0 00C0 0030 0060   EMULIN - 1 = CPCR     % (Y") INTO B                 05354000 D
1C51  49C9 0C41 0B3C 00F0   B L = B; SET LC1      % LEFT JUSTIFY (Y")           05355000 D
1C52  0000 00C0 0030 0020   COMP 16 = SAR                                       05356000 D
1C53  240C 00C0 0030 C060   DIV - 1 = CPCR        % (R(A),R(A+1))/(Y")          05357000 D
1C54  2BCB 00C0 0000 00F0   IF LC1 THEN SET LC1; STEP ELSE SKIP  % CHECK FOR OVERFLOW 05358000 D
1C55  50AC 00C0 0000 0E60   SETOVBIT - 1 = CPCR   % SET THE OV BIT               05359000 D
1C56  2819 00C0 0030 00FC   IF LC1 THEN SKIP                                    05360000 D
1C57  5050 00C0 0000 0060   CLEAROV - 1 = CPCR    % CLEAR THE OV BIT            05361000 D
1C58  4809 00C0 0030 00FC   A2 = MIR              % REMAINDER INTO MIR           05362000 D
1C59  4809 E0C0 0B3C 00F0   A3 = B                % QUOTIENT INTO B              05363000 D
1C5A  2000 00C0 0000 0060   SETCCA - 1 = CPCR     % SET THE CONDITION BITS       05364000 D
1C5B  4809 00C0 0000 20FC   ASR                   % REF OR1                      05365000 D
1C5C  4809 0F43 801C 00F0   BMAR R = MAR2         % R(A)                         05366000 D
1C5D  0000 00C0 0000 C01C   B = SAR                                             05367000 D
1C5E  2F50 00C0 0030 0060   EOUTPUT - 1 = CPCR                                  05368000 D
1C5F  4809 E0C0 0030 00F0   A3 = MIR                                            05369000 D
1C60  4809 0F46 0B30 00F0   BMAR + 1 = B          % R(A+1)                      05370000 D
1C61  51C0 0000 0030 C060   REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2            05371000 D
1C62  2F50 00C0 0030 0060   EOUTPUT - 1 = CPCR                                  05372000 D
1C63  66E0 00C0 0030 0040   OPCODE - 1 = MPCR                                   05373000 D
                                                  %                             05374000 D
                                                  %                             05375000 D
                      OP272:                      % TYPE RK DIVIDE (CONSTANT)    05376000 D
                                                  % (R(A),R(A+1))/Y = R(A+1), REMAINDER 05377000 D
                                                  % INTO R(A), SET CC AND OV     05378000 D
1C64  4809 E156 0B00 00F0   A3 AND LIT = B        % ISOLATE "M"                 05379000 D
1C65  00F0 0000 0000 00E0   15 = LIT                                            05380000 D
1C66  4809 CC52 0000 00F0   0 EQL B               % CHECK FOR M = 0             05381000 D
1C67  6819 00C0 0030 00F0   IF TRUE THEN SKIP                                   05382000 D
1C68  2380 00C0 0000 0060   CONTENTSRM - 1 = CPCR  % (R(M)) INTO B               05383000 D
1C69  4809 0C41 0B3C 00FC   B L = B                                             05384000 D
1C6A  0000 0000 0030 C020   COMP 16 = SAR                                       05385000 D
1C6B  48C9 EC50 1000 00FC   A3 OR B = A3          % A3 = (R(M))/INSTRUCTION     05386000 D
1C6C  6330 00C0 0000 C060   IFETCH - 1 = CPCR     % "Y" INTO B                 05387000 D
1C6D  4809 E0C0 A000 C0F0   A3 R = A2             % (R(M)) INTO A2              05388000 D
1C6E  0000 00C0 0030 C020   16 = SAR                                            05389000 D
1C6F  4809 CC41 0B0C 00FC   A2 + B L = B          % Y INTO HI WORD OF B         05390000 D
1C70  4809 E0C1 1030 C0F0   A3 L = A3             % CLEAR UHW OF A3             05391000 D
1C71  4809 E003 9030 C0F0   A3 R = A3                                           05392000 D
1C72  4809 EC50 1000 00FC   A3 OR B = A3          % A3 = Y/INSTRUCTION          05393000 D
1C73  4809 E0C0 8B00 00F0   A3 R = B                                            05394000 D
1C74  30FC 00C0 0000 008C   4 = SAR; 15 = LIT                                   05395000 D
1C75  4809 2C56 0B00 00F0   LIT AND B = B         % ISOLATE "A"                 05396000 D
1C76  51C0 0000 0030 0060   REGSTACK - 1 = CPCR                                 05397000 D
```

```
1C77  4809 0F41 0020 00F0          BMAR L = BR1          % TEMP STORAGE                                  05398000 D
1C78  00CC 0000 0000 0030          COMP 8 = SAR                                                          05399000 D
1C79  2F30 0000 0000 0060          EINPUT - 1 = CPCR     % (R(A)) INTO E                                 05400000 D
1C7A  4809 0C41 2000 00F0          B L = A2                                                              05401000 D
1C7B  0000 C000 0030 002C          COMP 16 = SAR                                                         05402000 D
1C7C  4809 0F46 0B0C 00F0          BMAR + 1 = B          % R(A+1)                                        05403000 D
1C7D  51CC 0000 0000 0060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2                              05404000 D
1C7E  2F30 0000 0000 0060          EINPUT - 1 = CPCR     % (R(A+1)) INTO B                               05405000 D
1C7F  4809 CC50 2030 00F0          A2 OR B = A2          % A2 = (R(A))/(R(A+1))                           05406000 D
1C80  49C9 E000 8B00 00F0          A3 R = B; SET LC1     % Y INTO B                                       05407000 D
1C81  0000 0000 0000 0020          16 = SAR                                                              05408000 D
1C82  4809 0C41 0B0C 00F0          B L = B               % LEFT JUSTIFY DIVISOR                           05409000 D
1C83  2400 0000 0000 0060          DIV - 1 = CPCR        % (R(A),R(A+1))/Y                                05410000 D
1C84  28CB 0000 0000 00F0          IF LC1 THEN SET LC1; STEP ELSE SKIP  % CHECK FOR OVERFLOW              05411000 D
1C85  50AC 0000 0000 0060          SETOVBIT - 1 = CPCR   % SET THE OV BIT                                05412000 D
1C86  2819 0000 0000 00F0          IF LC1 THEN SKIP                                                      05413000 D
1C87  5050 0000 0000 0060          CLEAROV - 1 = CPCR    % CLEAR THE OV BIT                              05414000 D
1C88  4809 E000 0B00 00F0          A3 = B                % QUOTIENT FOR SET CC                            05415000 D
1C89  2600 0000 0000 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS                         05416000 D
1C8A  4809 C000 0030 20F0          A2 = MIR; ASR         % REMAINDER INTO MIR                             05417000 D
1C8B  48C9 0F40 8C1C 00F0          BMAR R = MAR2         % R(A)                                           05418000 D
1C8C  0000 0000 0000 0010          6 = SAR                                                               05419000 D
1C8D  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                                    05420000 D
1C8E  4809 0F46 0B30 00F0          BMAR + 1 = B          % P(A+1)                                        05421000 D
1C8F  51C0 0000 0000 0060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2                              05422000 D
1C90  4809 E000 0030 00F0          A3 = MIR              % QUOTIENT INTO MIR                              05423000 D
1C91  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR                                                    05424000 D
1C92  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                                     05425000 D
                                                         %                                                 05426000 D
                                                         %                                                 05427000 D
                                OP273:                   % TYPE RX DIVIDE, (R(A),R(A+1))/(Y) =            05428000 D
                                                         % R(A+1), REMAINDER INTO R(A), SET OV, CC05429000 D
1C93  570C 0000 0000 0060          RXMFIELD - 1 = CPCR   % Y INTO B                                      05430000 D
1C94  4809 0C41 0010 00F0          B L = BR2                                                             05431000 D
1C95  0000 C000 0030 0030          COMP 8 = SAR                                                          05432000 D
1C96  60E0 0000 0000 0060          EMULIN - 1 = CPCR     % (Y) INTO B                                    05433000 D
1C97  4809 CC40 0030 00F0          B = MIR                                                               05434000 D
1C98  4809 E000 8B00 00F0          A3 R = B                                                              05435000 D
1C99  50F0 0000 0000 0080          4 = SAR; 15 = LIT                                                     05436000 D
1C9A  48C9 2C56 0B0C 00F0          LIT AND B = B         % ISOLATE "A"                                   05437000 D
1C9B  51CC 0000 0000 0060          REGSTACK - 1 = CPCR   % R(A) INTO MAR2                                05438000 D
1C9C  4809 0F41 0020 00F0          BMAR L = BR1          % TEMP STORAGE                                  05439000 D
1C9D  0000 0000 0000 0030          COMP 8 = SAR                                                          05440000 D
1C9E  2F30 0000 0000 0060          EINPUT - 1 = CPCR     % (R(A)) INTO E                                 05441000 D
1C9F  4809 0C41 2000 00F0          B L = A2                                                              05442000 D
1CA0  0000 0000 0000 0020          COMP 16 = SAR                                                         05443000 D
1CA1  4809 0F46 0B0C 00F0          BMAR + 1 = B          % R(A+1)                                        05444000 D
1CA2  51CC 0000 0000 0060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2                              05445000 D
1CA3  2F30 0000 0000 0060          EINPUT - 1 = CPCR     % (R(A+1)) INTO B                               05446000 D
1CA4  4809 CC50 2030 00F0          A2 OR B = A2          % A2 = (R(A))/(R(A+1))                           05447000 D
1CA5  49C9 0000 0000 00F0          BMI; SET LC1                                                          05448000 D
1CA6  4809 0C41 0B30 00F0          B L = B               % LEFT JUSTIFY THE DIVISOR                       05449000 D
1CA7  0000 0000 0000 0020          COMP 16 = SAR                                                         05450000 D
1CA8  2400 0000 0000 0060          DIV - 1 = CPCR        % (R(A),R(A+1))/(Y)                             05451000 D
1CA9  28CB 0000 0000 00F0          IF LC1 THEN SET LC1; STEP ELSE SKIP  % CHECK FOR OVERFLOW              05452000 D
1CAA  50AC 0000 0000 0060          SETOVBIT - 1 = CPCR   % SET THE OV BIT                                05453000 D
1CAB  2819 0000 0000 00F0          IF LC1 THEN SKIP                                                      05454000 D
1CAC  5050 0000 0000 0060          CLEAROV - 1 = CPCR    % CLEAR THE OV BIT                              05455000 D
1CAD  4809 E000 0B30 00F0          A3 = B                % QUOTIENT INTO B                                05456000 D
1CAE  2000 0000 0000 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS                         05457000 D
```

```
1C4F  4809 0000 0030 20F0          A2 = MIR; ASR          % REMAINDER INTO MIR          05458000 D
1C80  4809 0F43 801C 00F0          BMAR R = MAR2          % R(A)                        05459000 D
1C81  0000 0000 0030 0010          B = SAR                                              05460000 D
1C82  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR     % QUOTIENT INTO R(A+1)        05461000 D
1C83  4809 0F46 0830 00F0          BMAR + 1 = B           % R(A+1)                      05462000 D
1C84  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2            05463000 D
1C85  4809 E003 0090 00FC          A3 = MIR               % QUOTIENT INTO MIR           05464000 D
1C86  2F50 0000 0000 C060          EOUTPUT - 1 = CPCR                                   05465000 D
1C87  66E0 0000 0030 C040          OPCODE - 1 = MPCR                                    05466000 D
                                                          %                             05467000 D
                                                          %                             05468000 D
1C88  5F90 0000 0030 0060  OPCODE30:  XECODE - 1 = CPCR   % "F" INTO A2                 05469000 D
1C89  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                   05470000 D
1C8A  18EC 0000 0030 C000          OP30F - 1 = AMPCR                                    05471000 D
1C8B  4809 0000 0000 00FC          STEP                                                 05472000 D
1C8C  4824 0000 0000 00FC          EXEC                                                 05473000 D
                                                          %                             05474000 D
1C8D  18FC 0000 0030 0040  OP30F:  OP300 - 1 = MPCR                                     05475000 D
1C8E  1900 0000 0030 C040          OP301 - 1 = MPCR                                     05476000 D
1C8F  191C 0000 0000 004C          OP302 - 1 = MPCR                                     05477000 D
1C90  1920 0000 0000 0040          OP303 - 1 = MPCR                                     05478000 D
                                                          %                             05479000 D
                                                          %                             05480000 D
                           OP300:                         % TYPE RR AND (REGISTER)      05481000 D
                                                          % (R(A)) AND (R(M)) = R(A), SET CC 05482000 D
1C91  228C 0000 0030 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B               05483000 D
1C92  4809 0C40 0030 00FC          B = MIR                % USED AS TEMP                 05484000 D
1C93  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B               05485000 D
1C94  4809 0C40 2000 00FC          B = A2, BMI                                          05486000 D
1C95  4809 CC56 0890 C0F0          A2 AND B = MIR, B                                    05487000 D
1C96  4809 E003 801C 00FC          A3 R = MAR2            % R(A)                        05488000 D
1C97  0000 0000 0000 0020          16 = SAR                                             05489000 D
1C98  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                   05490000 D
1C99  2000 0000 0000 C060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS      05491000 D
1C9A  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                    05492000 D
                                                          %                             05493000 D
                                                          %                             05494000 D
                           OP301:                         % RI TYPE 2 AND (INDIRECT)     05495000 D
                                                          % (R(A)) AND (Y*) = R(A), SET CC 05496000 D
1C9B  2380 0000 0030 0060          CONTENTSRM - 1 = CPCR  % Y* INTO B                   05497000 D
1C9C  4809 0C41 0010 00F0          B L = BR2                                            05498000 D
1C9D  0000 0000 0000 0030          COMP B = SAR                                         05499000 D
1C9E  6CE0 0000 0000 0060          EMULIN - 1 = CPCR      % (Y*) INTO B                 05500000 D
1C9F  4809 0C40 0030 00F0          B = MIR                                              05501000 D
1CA0  4809 E003 0800 00F0          A3 = B                                               05502000 D
1CA1  228C 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B               05503000 D
1CA2  4809 0C40 2000 00F0          B = A2,BMI                                           05504000 D
1CA3  4809 CC56 0890 00FC          A2 AND B = MIR, B                                    05505000 D
1CA4  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                   05506000 D
1CA5  2000 0000 0030 006C          SETCCA - 1 = CPCR      % SET THE CONDITION BITS      05507000 D
1CA6  66E0 0000 0030 0040          OPCODE - 1 = MPCR                                    05508000 D
                                                          %                             05509000 D
                                                          %                             05510000 D
                           OP302:                         % TYPE RK AND (CONSTANT)      05511000 D
                                                          % (R(A)) AND Y = R(A), SET CC 05512000 D
1CA7  4809 2C56 0800 00F0          LIT AND B = B          % ISOLATE "M"                 05513000 D
1CA8  00FC 0000 0030 C0EC          15 = LIT                                             05514000 D
1CA9  4809 0C52 0000 00F0          0 EQL B                % CHECK FOR M = 0              05515000 D
1CAA  6819 0000 0000 00F0          IF TRUE THEN SKIP                                    05516000 D
1CAB  238C 0000 0000 C060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B               05517000 D
```

```
1CDC  4809 0C41 0B0C C0F0          B L = B                                          05518C00 D
1CDD  0000 0000 0000 0020          COMP 16 = SAR                                    05519C00 D
1CDE  4809 EC5C 1030 C0F0          A3 OR B = A3          % A3 = (R(M))/INSTRUCTION  05520C00 D
1CDF  633C 0000 0000 0060          IFETCH - 1 = CPCR     % "Y" INTO B              05521C00 D
1CE0  4809 E000 AC00 00FC          A3 R = A2                                        05522C00 D
1CE1  0000 0000 0000 0020          16 = SAR                                         05523C00 D
1CE2  4809 0C40 0030 00F0          A2 + B = MIR          % Y INTO MIR               05524C00 D
1CE3  4809 E000 0B30 00FC          A3 = B                                           05525C00 D
1CE4  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B          05526C00 D
1CE5  4809 0C40 2D00 00F0          B = A2, BMI                                      05527C00 D
1CE6  4809 CC56 0B30 00F0          A2 AND B = MIR, B                                05528C00 D
1CE7  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                               05529C00 D
1CE8  2000 0000 0000 C06C          SETCCA - 1 = CPCR     % SET THE CONDITION BITS   05530C00 D
1CE9  66EC 0000 0030 C040          OPCODE - 1 = MPCR                                05531C00 D
                                                         %                          05532C00 D
                                                         %                          05533C00 D
                                   OP3C3:                % RX TYPE AND               05534C00 D
                                                         % (R(A)) AND (Y) = R(A), SET CC 05535C00 D
1CEA  570C 0000 0000 006C          PXMFIELD - 1 = CPCR   % Y INTO B                 05536C00 D
1CEB  4809 0C41 0C10 C0F0          B L = BR2                                        05537C00 D
1CEC  0000 0000 0000 C030          COMP 8 = SAR                                     05538C00 D
1CED  60E0 0000 0030 0060          EMULIN - 1 = CPCR     % (Y) INTO B               05539C00 D
1CEE  4809 0C40 0030 00F0          B = MIR                                          05540C00 D
1CEF  4809 E000 0B00 00F0          A3 = B                                           05541C00 D
1CF0  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B           05542C00 D
1CF1  4809 0C40 2D00 00F0          B = A2, BMI                                      05543C00 D
1CF2  4809 CC56 0B30 00F0          A2 AND B = MIR, B                                05544C00 D
1CF3  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                               05545C00 D
1CF4  2000 0000 0000 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS   05546C00 D
1CF5  6EE0 0000 0000 0C40          OPCODE - 1 = MPCR                                05547C00 D
                                                         %                          05548C00 D
                                                         %                          05549C00 D
1CF6  5F90 0000 0000 C060          OPCODE31:  XFCODE - 1 = CPCR  % "F" INTO A2      05550C00 D
1CF7  4809 C640 0030 C0F0          A2 + AMPCR = AMPCR                               05551C00 D
1CF8  1930 0000 0000 00C0          OP31F - 1 = AMPCR                                05552C00 D
1CF9  4809 0000 0000 00F0          STEP                                             05553C00 D
1CFA  4824 0000 0000 00F0          EXEC                                             05554C00 D
                                                         %                          05555C00 D
1CFB  1940 0000 0000 0040          OP31F:  OP310 - 1 = MPCR                         05556C00 D
1CFC  1950 0000 0000 0040          OP311 - 1 = MPCR                                 05557C00 D
1CFD  1960 0000 0000 0040          OP312 - 1 = MPCR                                 05558C00 D
1CFE  1970 0000 0000 0040          OP313 - 1 = MPCR                                 05559C00 D
                                                         %                          05560C00 D
                                                         %                          05561C00 D
                                   OP310:                % RR TYPE OR (REGISTER)     05562C00 D
                                                         % (R(A)) OR (R(M)) = R(A), SET CC 05563C00 D
1CFF  2280 0000 0000 006C          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B           05564C00 D
1100  4809 0C40 0030 C0F0          B = MIR                                          05565C00 D
1101  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B           05566C00 D
1102  4809 0C40 2D00 00F0          B = A2, BMI                                      05567C00 D
1103  4809 CC50 0B30 00FC          A2 OR B = MIR, B                                 05568C00 D
1104  4809 E000 8010 C0FC          A3 R = MAR2           % R(A)                     05569C00 D
1105  0000 0000 0000 0020          16 = SAR                                         05570C00 D
1106  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                               05571C00 D
1107  2000 0000 0000 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS   05572C00 D
1108  66EC 0000 0000 C040          OPCODE - 1 = MPCR                                05573C00 D
                                                         %                          05574C00 D
                                                         %                          05575C00 D
                                   OP311:                % RI TYPE 2 OR (INDIRECT)   05576C00 D
                                                         % (R(A)) OR (Y*) = R(A), SET CC 05577C00 D
```

```
1109  2390 0000 0000 0060          CONTENTSRM - 1 = CPCR   % Y* INTO B                  05570000 D
110A  4809 0C41 0010 00F0          B L = BR2                                            05579000 D
110B  00FC 0000 0000 0030          COMP B = SAR                                         05580000 D
110C  50E0 0000 0000 006C          EMULIN - 1 = CPCR       % (Y*) INTO B                05581000 D
110D  4809 0C40 0030 00F0          B = MIR                                              05582000 D
110E  4809 E000 0B30 00F0          A3 = B                                               05583000 D
110F  2280 0000 0000 006C          CONTENTSRA - 1 = CPCR   % (R(A)) INTO B              05584000 D
1110  4809 0C40 2000 00FC          B = A2,BMI                                           05585000 D
1111  4809 CC5C 0B30 00F0          A2 OR B = MIR, B                                     05586000 D
1112  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                   05587000 D
1113  2000 0000 0000 006C          SETCCA - 1 = CPCR       % SET THE CONDITION BITS     05588000 D
1114  56E0 0000 0000 0040          OPCODE - 1 = MPCR                                    05589000 D
                                                           %                            05590000 D
                                                           %                            05591000 D
                                OP312:                     % RK TYPE OR (CONSTANT)       05592000 D
                                                           % (R(A)) OR Y = R(A), SET (C 05593000 D
1115  4809 2055 0B30 00F0          LIT AND B = B           % ISOLATE "M"                 05594000 D
1116  00FC 0000 0000 00E0          15 = LIT                                             05595000 D
1117  4809 0C52 0000 00F0          0 EQL B                 % CHECK FOR M = 0             05596000 D
1118  6819 0000 0000 00F0          IF TRUE THEN SKIP                                    05597000 D
1119  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR   % (R(M)) INTO B              05598000 D
111A  4809 0C41 0B00 00F0          B L = B                                              05599000 D
111B  0000 0000 0000 0020          COMP 16 = SAR                                        05600000 D
111C  4809 EC5C 1000 00F0          A3 OR B = A3            % A3 = (R(M))/INSTRUCTION     05601000 D
111D  6330 0000 0000 006C          IFETCH - 1 = CPCR       % "Y" INTO B                 05602000 D
111E  4809 E000 A000 00FC          A3 R = A2                                            05603000 D
111F  00FC 0000 0000 00A0          16 = SAR; 15 = LIT                                   05604000 D
1120  4809 CC40 0030 00F0          A2 + B = MIR            % Y INTO MIR                 05605000 D
1121  4809 E000 0B00 00F0          A3 = B                                               05606000 D
1122  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR   % (R(A)) INTO B              05607000 D
1123  4809 0C40 2000 00F0          B = A2, BMI                                          05608000 D
1124  4809 CC5C 0B30 00F0          A2 OR B = MIR, B                                     05609000 D
1125  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                   05610000 D
1126  2000 0000 0000 006C          SETCCA - 1 = CPCR       % SET THE CONDITION BITS     05611000 D
1127  56E0 0000 0000 0040          OPCODE - 1 = MPCR                                    05612000 D
                                                           %                            05613000 D
                                                           %                            05614000 D
                                OP313:                     % RX TYPE OR                  05615000 D
                                                           % (R(A)) OR (Y) = R(A), SET CC 05616000 D
1128  5700 0000 0000 0060          RXMFIELD - 1 = CPCR     % Y INTO B                   05617000 D
1129  4809 0C41 0010 00F0          B L = BR2                                            05618000 D
112A  00FC 0000 0000 0030          COMP B = SAR                                         05619000 D
112B  50E0 0000 0000 0060          EMULIN - 1 = CPCR       % (Y) INTO B                 05620000 D
112C  4809 0C40 0030 00F0          B = MIR                                              05621000 D
112D  4809 E000 0B00 00F0          A3 = B                                               05622000 D
112E  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR   % (R(A)) INTO B              05623000 D
112F  4809 0C40 2000 00F0          B = A2, BMI                                          05624000 D
1130  4809 CC5C 0B30 00F0          A2 OR B = MIR, B                                     05625000 D
1131  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                   05626000 D
1132  2000 0000 0000 0060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS     05627000 D
1133  56E0 0000 0000 0040          OPCODE - 1 = MPCR                                    05628000 D
                                                           %                            05629000 D
                                                           %                            05630000 D
1134  5F90 0000 0000 0060     OPCODE32:  XFCODE - 1 = CPCR % "F" CODE INTO A2           05631000 D
1135  4809 0640 0030 00F0          A2 + AMPCR = AMPCR                                   05632000 D
1136  1980 0000 0000 00C0          OP32F - 1 = AMPCR                                    05633000 D
1137  4809 00C0 0000 00F0          STEP                                                 05634000 D
1138  4824 0000 0000 00F0          EXEC                                                 05635000 D
                                                           %                            05636000 D
1139  1990 0000 0000 0040     OP32F:  OP32C - 1 = MPCR                                  05637000 D
```

```
113A  19A0 0000 0000 0040          OP321 - 1 = MPCR
113B  19B0 0000 0000 C040          OP322 - 1 = MPCR
113C  19C0 0000 0000 C04C          OP323 - 1 = MPCR
                                                   %
                                                   %
                                 OP320:            % RR TYPE EXCLUSIVE OR (REGISTER)
                                                   % (R(A)) XOR (R(M)) = R(A), SET CC
113D  2280 C000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B
113E  9809 0C40 0030 00F0          B = MIR
113F  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B
1140  9809 0C40 2000 00FC          B = A2, BMI
1141  9809 CC4C 0030 00F0          A2 XOR B = MIR, B
1142  9809 E000 801C 00FC          A3 R = MAR2            % R(A)
1143  0000 0000 0000 0020          16 = SAR
1144  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR
1145  2000 0000 0030 006C          SETCCA - 1 = CPCR      % SET THE CONDITION BITS
1146  56E0 0000 0000 0040          OPCODE - 1 = MPCR
                                                   %
                                                   %
                                 OP321:            % RI TYPE 2 EXCLUSIVE OR (INDIRECT)
                                                   % (R(A)) XOR (Y*) = R(A), SET CC
1147  2380 0000 0000 C060          CONTENTSRM - 1 = CPCR  % Y* INTO B
1148  9809 0C41 001C C0F0          B L = BR2
1149  0000 0000 0000 0030          COMP B = SAR
114A  50E0 0000 0000 006C          EMULIN - 1 = CPCR      % (Y*) INTO B
114B  9809 CC40 0030 00FC          B = MIR
114C  9809 E0F0 0B30 00F0          A3 = B
114D  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B
114E  9809 0C40 2000 00F0          B = A2,BMI
114F  9809 CC4C 0B30 00F0          A2 XOR B = MIR, B
1150  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR
1151  2000 0000 0030 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS
1152  56E0 0000 0000 C040          OPCODE - 1 = MPCR
                                                   %
                                                   %
                                 OP322:            % RK TYPE EXCLUSIVE OR (CONSTANT)
                                                   % (R(A)) XOR Y = R(A), SET CC
1153  9809 2C56 0B30 00FC          LIT AND B = B          % ISOLATE "M"
1154  00F0 0000 0030 C0E0          15 = LIT
1155  9809 0C52 0000 00F0          0 EQL B                % CHECK FOR M = 0
1156  5819 0000 0000 C0F0          IF TRUE THEN SKIP
1157  2380 0000 0000 C060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B
1158  9809 0C41 0B30 C0F0          B L = B
1159  0000 0000 0030 0020          COMP 16 = SAR
115A  9809 E05C 1000 00FC          A3 OR B = A3           % A3 = (R(M))/INSTRUCTION
115B  5330 0000 0000 0060          IFETCH - 1 = CPCR      % "Y" INTO B
115C  9809 E0F0 A030 C0F0          A3 R = A2
115D  00F0 0000 0030 00A0          16 = SAR; 15 = LIT
115E  9809 CC40 0030 C0F0          A2 + B = MIR           % Y INTO MIR
115F  9809 E0F0 0B30 00FC          A3 = B
1160  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B
1161  9809 0C40 2000 00FC          B = A2, BMI
1162  9809 CC4C 0B30 C0F0          A2 XOR B = MIR, B
1163  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR
1164  2000 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS
1165  56E0 0000 0000 C040          OPCODE - 1 = MPCR
                                                   %
                                                   %
                                 OP323:            % RX TYPE EXCLUSIVE OR
                                                   % (R(A)) XOR (Y) = R(A), SET CC
```

```
1166  5700 0000 0000 0060          RXMFIELD - 1 = CPCR    % Y INTO B                          05698000 D
1167  4809 0041 0010 00F0          B L = BR2                                                  05699000 D
1168  0000 0000 0000 0030          COMP B = SAR                                               05700000 D
1169  60EC 0000 0000 0060          EMULIN - 1 = CPCR      % (Y) INTO B                        05701000 D
116A  4809 0040 0090 00F0          B = MIR                                                    05702000 D
116B  4809 E000 0B00 00F0          A3 = B                                                     05703000 D
116C  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                     05704000 D
116D  4809 0040 2000 00F0          B = A2, BMI                                                05705000 D
116E  4809 0040 0B30 00F0          A2 XOR B = MIR, B                                          05706000 D
116F  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                         05707000 D
1170  2000 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS            05708000 D
1171  66F0 0000 0000 0040          OPCODE - 1 = MPCR                                          05709000 D
                                                          %                                   05710000 D
                                                          %                                   05711000 D
1172  5F90 0000 0000 0060    OPCODE33:  XFCODE - 1 = CPCR % "F" INTO A2                       05712000 D
1173  4809 0640 0040 00F0          A2 + AMPCR = AMPCR                                         05713000 D
1174  19D0 0000 0000 0000          OP33F - 1 = AMPCR                                          05714000 D
1175  4809 0000 0000 00F0          STEP                                                       05715000 D
1176  4824 0000 0000 00F0          EXEC                                                       05716000 D
                                                          %                                   05717000 D
1177  19E0 0000 0000 0040    OP33F:  OP330 - 1 = MPCR                                         05718000 D
1178  19F0 0000 0000 0040          OP331 - 1 = MPCR                                           05719000 D
1179  1A00 0000 0000 0040          OP332 - 1 = MPCR                                           05720000 D
117A  1A10 0000 0000 0040          OP333 - 1 = MPCR                                           05721000 D
                             OP330:                       % TYPE RR MASKED SUBSTITUTE (REGISTER)  05722000 D
                                                          % IF (R(A+1))N = 1, (R(M)) INTO R(A)N, CC05723000 D
117B  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                     05724000 D
117C  4809 0040 0090 00F0          B = MIR                % TEMP STORAGE                       05725000 D
117D  4809 E000 8B30 00F0          A3 R = B                                                   05726000 D
117E  30F0 0000 0000 0080          4 = SAR, 15 = LIT                                          05727000 D
117F  4809 2056 0000 00F0          LIT AND B = B          % ISOLATE "A"                       05728000 D
1180  5100 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A) INTO MAR2                    05729000 D
1181  4809 0F41 0020 00F0          BMAR L = BR1           % STORE R(A)                        05730000 D
1182  0000 0000 0000 0030          COMP B = SAR                                               05731000 D
1183  4809 0F46 0B00 00F0          BMAR + 1 = B           % R(A+1)                            05732000 D
1184  5100 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2                  05733000 D
1185  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B                   05734000 D
1186  4809 0040 2000 00F0          B = A2, BMI                                                05735000 D
1187  4809 0056 0030 00F0          A2 AND B = MIR         % (R(A+1)) AND (R(M))               05736000 D
1188  4809 0000 0000 20F0          ASR                    % REFERENCE BR1                     05737000 D
1189  4809 0F40 801C 00F0          BMAR R = MAR2                                              05738000 D
118A  0000 0000 0000 0010          B = SAR                                                    05739000 D
118B  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO P                     05740000 D
118C  4809 0040 2000 00F0          B = A2,BMI                                                 05741000 D
118D  4809 0050 0B30 00F0          A2 OR B = MIR, B                                           05742000 D
118E  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                         05743000 D
118F  2000 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS            05744000 D
1190  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                          05745000 D
                                                          %                                   05746000 D
                                                          %                                   05747000 D
                             OP331:                       % RI TYPE 2 MASKED SUBSTITUTE (INDIRECT) 05748000 D
                                                          % IF (R(A+1))N=1, (Y*)N = R(A)N, SET CC 05749000 D
1191  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                     05750000 D
1192  4809 0041 0010 00F0          B L = BR2                                                  05751000 D
1193  0000 0000 0000 0030          COMP B = SAR                                               05752000 D
1194  60EC 0000 0000 0060          EMULIN - 1 = CPCR      % (Y*) INTO B                       05753000 D
1195  4809 0040 0090 00F0          B = MIR                % TEMP STORAGE                       05754000 D
1196  4809 E000 8B30 00F0          A3 R = B                                                   05755000 D
1197  30F0 0000 0000 0080          4 = SAR, 15 = LIT                                          05756000 D
1198  4809 2056 0B30 00F0          LIT AND B = B          % ISOLATE "A"                       05757000 D
```

```
11B9  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A) INTO MAR2                    05758000 D
11BA  4809 0F41 0020 00FC          BMAR L = BR1           % TEMP STORAGE                       05759000 D
11BB  0000 0000 0030 0030          COMP B = SAR                                                05760000 D
11BC  4809 0F46 0B30 00F0          BMAR + 1 = B           % R(A+1)                             05761000 D
11BD  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2                  05762000 D
11BE  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B                   05763000 D
11BF  4809 0C40 2000 00F0          B = A2, BMI                                                 05764000 D
11C0  4809 0C56 0090 00F0          A2 AND B = MIR         % (R(A+1)) AND (Y*)                 05765000 D
11C1  4809 0000 0000 20F0          ASR                    % REFERENCE BR1                     05766000 D
11C2  4809 0F43 801C 00F0          BMAR R = MAR2          % R(A)                              05767000 D
11C3  0000 0000 0000 001C          B = SAR                                                     05768000 D
11C4  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B                     05769000 D
11C5  4809 0C40 2030 00F0          B = A2,BMI                                                  05770000 D
11C6  4809 0C50 0B90 00F0          A2 OR B = MIR, B                                            05771000 D
11C7  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                          05772000 D
11C8  2D0C 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS            05773000 D
11C9  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                           05774000 D
                                                          %                                   05775000 D
                                                          %                                   05776000 D
                                   OP332:                 % RK TYPE MASKED SUBSTITUTE (CONSTANT) 05777000 D
                                                          % IF (R(A+1))N= 1, YN INTO R(A)N, SET CC 05778000 D
11CA  4809 2C56 0B30 00F0          LIT AND B = B          % ISOLATE "M"                        05779000 D
11CB  00FC 0000 0030 00E0          15 = LIT                                                    05780000 D
11CC  4809 0C52 0020 00F0          0 EQL B                % CHECK FOR M = 0                    05781000 D
11CD  6819 0000 0000 00FC          IF TRUE THEN SKIP                                           05782000 D
11CE  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                     05783000 D
11CF  4809 0C41 0B30 00F0          B L = B                                                     05784000 D
11D0  0000 0000 0030 0020          COMP 16 = SAR                                               05785000 D
11D1  4809 EC5C 1000 00F0          A3 OR B = A3           % A3 = (R(M))/INSTRUCTION           05786000 D
11D2  633C 0000 0030 0060          IFETCH - 1 = CPCR      % "Y" INTO B                        05787000 D
11D3  4809 E000 A000 00F0          A3 R = A2              % (R(M)) INTO A2                    05788000 D
11D4  0000 0000 0000 0020          16 = SAR                                                    05789000 D
11D5  4809 0C40 0B30 00F0          A2 + B = MIR           % Y INTO MIR                        05790000 D
11D6  4809 E000 8800 00F0          A3 R = B                                                    05791000 D
11D7  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                           05792000 D
11D8  4809 2C56 0B00 00F0          LIT AND B = B          % ISOLATE "A"                       05793000 D
11D9  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A) INTO MAR2                    05794000 D
11DA  4809 0F41 0020 00FC          BMAR L = BR1           % TEMP STORAGE                       05795000 D
11DB  0000 0000 0030 0030          COMP B = SAR                                                05796000 D
11DC  4809 0F46 0B30 00F0          BMAR + 1 = B           % R(A+1)                             05797000 D
11DD  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2                  05798000 D
11DE  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B                   05799000 D
11DF  4809 0C40 2000 00F0          B = A2, BMI                                                 05800000 D
11E0  4809 0C56 0030 00F0          A2 AND B = MIR         % (R(A+1)) AND Y                    05801000 D
11E1  4809 0000 0000 20F0          ASR                    % REFERENCE BR1                     05802000 D
11E2  4809 0F43 801C 00F0          BMAR R = MAR2          % R(A)                              05803000 D
11E3  0000 0000 0000 0010          B = SAR                                                     05804000 D
11E4  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B                     05805000 D
11E5  4809 0C40 2000 00F0          B = A2,BMI                                                  05806000 D
11E6  4809 0C50 0B90 00F0          A2 OR B = MIR, B                                            05807000 D
11E7  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                          05808000 D
11E8  2D0C 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS            05809000 D
11E9  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                           05810000 D
                                                          %                                   05811000 D
                                                          %                                   05812000 D
                                   OP333:                 % RX TYPE MASKED SUBSTITUTE          05813000 D
                                                          % IF (R(A+1))N= 1, (Y)N INTO R(A), SET CC 05814000 D
11EA  570C 0000 0000 0060          RXMFIELD - 1 = CPCR    % Y INTO B                          05815000 D
11EB  4809 0C41 0010 00F0          B L = BR2                                                   05816000 D
11EC  0000 0000 0000 0030          COMP B = SAR                                                05817000 D
```

```
11CD  60E0 0000 0000 0060          EMULIN - 1 = CPCR      % (Y) INTO B                    05818000 D
11CE  4809 0C40 0090 00F0          B = MIR                % TEMP STORAGE                   05819000 D
11CF  4809 E000 8B00 00F0          A3 R = B                                               05820000 D
11D0  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                      05821000 D
11D1  4809 2056 0000 00F0          LIT AND B = B          % ISOLATE "A"                   05822000 D
11D2  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A) INTO MAR2                05823000 D
11D3  4809 0F41 0020 00F0          BMAR L = BR1           % TEMP STORAGE                   05824000 D
11D4  0000 00C0 0000 0030          COMP B = SAR                                           05825000 D
11D5  4809 0F46 0B00 00F0          BMAR + 1 = B           % R(A+1)                         05826000 D
11D6  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2              05827000 D
11D7  2F30 0000 0030 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B               05828000 D
11D8  4809 0C40 2000 00F0          B = A2, BMI                                            05829000 D
11D9  4809 0056 0030 00F0          A2 AND B = MIR         % (R(A+1)) AND (Y)              05830000 D
11DA  4809 0000 0000 20F0          ASR                    % REFERENCE BR1                 05831000 D
11DB  4809 0F40 8010 00F0          BMAR R = MAR2                                          05832000 D
11DC  0000 00C0 0000 0010          B = SAR                                                05833000 D
11DD  2F30 0000 0030 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B                 05834000 D
11DE  4809 0C40 2000 00F0          B = A2,BMI                                             05835000 D
11DF  4809 0050 0890 00F0          A2 OR B = MIR, B                                       05836000 D
11E0  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                     05837000 D
11E1  2000 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS        05838000 D
11E2  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                      05839000 D
                                                          %                               05840000 D
                                                          %                               05841000 D
11E3  5F90 0000 0000 0060  OPCODE34:  XFCODE - 1 = CPCR   % "F" INTO A2                   05842000 D
11E4  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                     05843000 D
11E5  1A20 0000 0000 00C0          OP34F - 1 = AMPCR                                      05844000 D
11E6  4809 0000 0000 00F0          STEP                                                   05845000 D
11E7  4824 0000 0000 00F0          EXEC                                                   05846000 D
                                                          %                               05847000 D
11E8  1A30 0000 0030 0040  OP34F:  OP340 - 1 = MPCR                                       05848000 D
11E9  1A40 0000 0030 0040          OP341 - 1 = MPCR                                       05849000 D
11EA  1A50 0000 0030 0040          OP342 - 1 = MPCR                                       05850000 D
11EB  1A60 0000 0030 0040          OP343 - 1 = MPCR                                       05851000 D
                                                          %                               05852000 D
                                                          %                               05853000 D
                           OP340:                         % TYPE RR COMPARE MASKED (REGISTER)  05854000 D
                                                          % (R(A)) AND (R(A+1)) : (R(M)) AND   05855000 D
                                                          % (R(A+1)), SET CC               05856000 D
11EC  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                 05857000 D
11ED  4809 0C40 0090 00F0          B = MIR                % TEMP STORAGE                   05858000 D
11EE  4809 E000 8B00 00F0          A3 R = B                                               05859000 D
11EF  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                      05860000 D
11F0  4809 2056 0B00 00F0          LIT AND B = B          % ISOLATE "A"                   05861000 D
11F1  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                                    05862000 D
11F2  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B                 05863000 D
11F3  4809 0C40 2000 00F0          B = A2                                                 05864000 D
11F4  4809 0F46 0B00 00F0          BMAR + 1 = B           % R(A+1)                         05865000 D
11F5  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2              05866000 D
11F6  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B               05867000 D
11F7  4809 0056 2000 00F0          A2 AND B = A2          % (R(A)) AND (R(A+1))           05868000 D
11F8  4809 0C40 1000 00F0          B = A3, BMI                                            05869000 D
11F9  4809 E056 0B00 00F0          A3 AND B = B           % (R(M)) AND (R(A+1))           05870000 D
11FA  2000 0000 0030 0060          SETCC - 1 = CPCR       % SET THE CONDITION BITS        05871000 D
11FB  66E0 0000 0030 0040          OPCODE - 1 = MPCR                                      05872000 D
                                                          %                               05873000 D
                                                          %                               05874000 D
                           OP341:                         % RI TYPE 2 COMPARE MASKED(INDIRECT)  05875000 D
                                                          % (R(A)) AND (R(A+1)) : (Y*) AND (R(A+1))05876000 D
                                                          % AND SET CC                     05877000 D
```

```
11FC  238C 0000 0030 0060        CONTENTSRM - 1 = CPCR  % Y* INTO B          05878C00 D
11FD  4809 0C41 0010 00F0        B L = BR2                                   05879C0C D
11FE  0C00 CC00 0030 0030        COMP B = SAR                                05880C00 D
11FF  60E0 0000 0000 0060        EMULIN - 1 = CPCR      % (Y*) INTO B        05881C00 D
1200  4809 0C40 0090 00FC        B = MIR                % TEMP STORAGE        05882C00 D
1201  4809 E000 8800 00F0        A3 R = B                                    05883C00 D
1202  80FC 0000 C00C C080        4 = SAR; 15 = LIT                           05884C00 D
1203  4809 2C56 0800 00FC        LIT AND B = B          % ISOLATE "A"        05885C0C D
1204  51C0 0000 0000 0060        REGSTACK - 1 = CPCR                         05886C00 D
1205  2F30 0000 0030 0060        EINPUT - 1 = CPCR      % (R(A)) INTO E      05887C00 D
1206  4809 0C40 2C00 00F0        B = A2                                      05888C0C D
1207  4809 0F45 0030 00F0        BMAR + 1 = B           % R(A+1)             05889C00 D
1208  51C0 0000 0000 0060        REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2   05890C00 D
1209  2F30 0000 0000 0060        EINPUT - 1 = CPCR      % (R(A+1)) INTO B    05891C00 D
120A  4809 CC56 2030 00FC        A2 AND B = A2          % (R(A)) AND (R(A+1)) 05892C0C D
120B  4809 0C40 1000 00F0        B = A3, BMI                                 05893C00 D
120C  4809 EC56 0800 00F0        A3 AND B = B           % (Y*) AND (R(A+1))  05894C00 D
120D  2000 0000 0000 0060        SETCC - 1 = CPCR       % SET THE CONDITION BITS 05895C00 D
120E  66E0 0000 0000 0040        OPCODE - 1 = MPCR                           05896C00 D
                                                        %                   05897C00 D
                                                        %                   05898C00 D
                          OP342:                        % TYPE RK COMPARE MASKED (CONSTANT) 05899C00 D
                                                        % (R(A)) AND (R(A+1)) : Y AND (R(A+1)), 05900C00 D
                                                        % AND SET CONDITION BITS 05901C00 D
120F  4809 2C56 0E00 00F0        LIT AND B = B          % ISOLATE M          05902C00 D
1210  00F0 0000 0000 00EC        15 = LIT                                    05903C00 D
1211  4809 0C52 0000 00F0        0 EQL B                % CHECK FOR M = 0     05904C00 D
1212  6819 0000 0000 00F0        IF TRUE THEN SKIP                           05905C00 D
1213  238C 0000 0030 C06C        CONTENTSRM - 1 = CPCR  % (R(M)) INTO B      05906C00 D
1214  4809 0C41 0800 00F0        B L = B                                     05907C00 D
1215  0000 0000 0030 0020        COMP 16 = SAR                               05908C00 D
1216  4809 EC5C 1000 00F0        A3 OR B = A3           % A3 = (R(M))/INSTRUCTION 05909C00 D
1217  5330 0000 0000 0060        IFETCH - 1 = CPCR      % "Y" INTO B         05910C00 D
1218  4809 EC00 A000 00F0        A3 R = A2              % (R(M)) INTO A2      05911C00 D
1219  0000 0000 0000 0020        16 = SAR                                    05912C00 D
121A  4809 0C40 0090 00F0        A2 + B = MIR           % Y INTO B           05913C00 D
121B  4809 E000 8800 00F0        A3 R = B                                    05914C00 D
121C  80FC 0000 0030 C080        4 = SAR; 15 = LIT                           05915C00 D
121D  4809 2C56 0800 00F0        LIT AND B = B          % ISOLATE "A"        05916C00 D
121E  51C0 0000 0000 0060        REGSTACK - 1 = CPCR    % R(A) INTO MAR2     05917C00 D
121F  2F30 0000 0000 0060        EINPUT - 1 = CPCR      % (R(A)) INTO E      05918C00 D
1220  4809 0C40 2000 00F0        B = A2                                      05919C00 D
1221  4809 0F45 0800 00FC        BMAR + 1 = B           % R(A+1)             05920C00 D
1222  51C0 0000 0000 0060        REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2   05921C00 D
1223  2F30 0000 0000 0060        EINPUT - 1 = CPCR      % (R(A+1)) INTO B    05922C00 D
1224  4809 CC56 2030 00F0        A2 AND B = A2          % (R(A)) AND (R(A+1)) 05923C00 D
1225  4809 0C40 1000 00F0        B = A3, BMI                                 05924C00 D
1226  4809 EC56 0800 00F0        A3 AND B = B           % Y AND (R(A+1))     05925C00 D
1227  2000 0000 0000 0060        SETCC - 1 = CPCR       % SET THE CONDITION BITS 05926C00 D
1228  66E0 0000 0000 0040        OPCODE - 1 = MPCR                           05927C00 D
                                                        %                   05928C00 D
                                                        %                   05929C00 D
                          OP343:                        % RX TYPE COMPARE MASKED 05930C00 D
                                                        % (R(A)) AND (R(A+1)) : (Y) AND (R(A+1)) 05931C00 D
                                                        % AND SET CONDITION BITS 05932C00 D
1229  570C 0000 0000 C06C        RXMFIELD - 1 = CPCR    % Y INTO B           05933C00 D
122A  4809 0C41 0010 00F0        B L = BR2                                   05934C00 D
122B  0000 0000 0000 C030        COMP B = SAR                                05935C00 D
122C  60E0 0000 0000 0060        EMULIN - 1 = CPCR      % (Y) INTO B         05936C00 D
122D  4809 0C40 0090 00FC        B = MIR                % TEMP STORAGE        05937C00 D
```

```
122E  4809 E000 8B30 00FC          A3 R = B                                  05938000 D
122F  80F0 0000 0000 0080          4 = SAR; 15 = LIT                         05939000 D
1230  4809 2056 0B00 00F0          LIT AND B = B          % ISOLATE "A"      05940000 D
1231  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A) INTO MAR2   05941000 D
1232  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO E    05942000 D
1233  4809 0C40 2000 00F0          B = A2                                    05943000 D
1234  4809 0F46 0B30 00FC          BMAR + 1 = B           % R(A+1)           05944000 D
1235  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2 05945000 D
1236  2F30 0000 0C20 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B  05946000 D
1237  4809 0C56 2000 00FC          A2 AND B = A2          % (R(A)) AND (R(A+1)) 05947000 D
1238  4809 0C40 1030 00F0          B = A3, BMI                              05948000 D
1239  4809 0C56 0F00 00F0          A3 AND B = B           % (Y) AND (R(A+1)) 05949000 D
123A  2000 0000 0C30 C060          SETCC - 1 = CPCR       % SET THE CONDITION BITS 05950000 D
123B  66F0 0000 0000 0040          OPCODE - 1 = MPCR                        05951000 D
                                                          %                 05952000 D
123C  5F90 0000 0000 0060  OPCODE35:  XFCODE - 1 = CPCR   % "F" INTO A2     05953000 D
123D  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                       05954000 D
123E  1A70 0000 0C30 C0C0          OP35F- 1 = AMPCR                         05955000 D
123F  4809 0000 0000 00F0          STEP                                     05956000 D
1240  4824 0000 0000 00F0          EXEC                                     05957000 D
                                                          %                 05958000 D
1241  1A80 0000 0000 0040  OP35F:  OP350 - 1 = MPCR                         05959000 D
1242  1A90 0000 0000 0040          OP351 - 1 = MPCR                         05960000 D
1243  1AA0 0000 0000 0040          OP352 - 1 = MPCR                         05961000 D
1244  1AB0 0000 0000 0040          OP353 - 1 = MPCR                         05962000 D
                                                          %                 05963000 D
                           OP350:                         % INPUT OUTPUT COMMAND WORD FETCH (5C) 05964000 D
                                                          % IOCW IS LOADED PRIOR TO EXECUTING THIS 05965000 D
                                                          % INSTRUCTION    05966000 D
1245  4809 2056 0B00 00F0          LIT AND B = B          % :M: AND :A: FIELDS MUST BE ZERO ELSE 05967000 D
                                                          % FAULT ENTERED. 05968000 D
1246  80F0 0000 0000 0080          15 = LIT; 4 = SAR                        05969000 D
1247  4809 0052 0000 00F0          0 NEQ B                                  05970000 D
1248  600B 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP              05971000 D
1249  3000 0000 0000 0040          FAULT - 1 = MPCR                         05972000 D
124A  4809 E000 8B30 00FC          A3 R = B                                 05973000 D
124B  4809 2056 0B00 00F0          LIT AND B = B                            05974000 D
124C  4809 0052 0030 00F0          0 NEQ B                                  05975000 D
124D  600B 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP              05976000 D
124E  3C0C 0000 0000 0040          FAULT - 1 = MPCR                         05977000 D
124F  4809 2000 0010 00F0          LIT = MAR2                               05978000 D
1250  A320 0000 0000 0080          IOCW = LIT; 6 = SAR                      05979000 D
1251  2F30 0000 0000 C060          EINPUT - 1 = CPCR      % B = (IOCW)      05980000 D
1252  4809 0C40 2000 00F0          B = A2                 % STORE (IOCW) IN A2 05981000 D
1253  4809 0C41 0B30 00FC          B L = B                % ISOLATE NO OF BUFFERS IN UHW OF B 05982000 D
1254  A000 0000 0000 001C          COMP 18 = SAR                            05983000 D
1255  4809 0C40 8B30 00F0          B R = B                                  05984000 D
1256  0000 0000 0000 0010          B = SAR                % SHIFT NO. OF BUFFERS IN UHW OF B 05985000 D
1257  4809 EC5C 1030 00F0          A3 OR B = A3           % STORE BUFFER COUNT IN UHW OF A3 05986000 D
1258  4809 C000 0030 00F0          A2                     % TEST BIT FOR I/O 05987000 D
1259  580B 0000 0000 00F0          IF LST THEN STEP ELSE SKIP               05988000 D
125A  367C 0000 0000 0040          EOUT - 1 = MPCR                          05989000 D
125B  3B40 0000 0000 C040          IN - 1 = MPCR                            05990000 D
                                                          %                 05991000 D
                                                          %                 05992000 D
                           OP351:                         % RI TYPE 2 BIASED FETCH 05993000 D
                                                          % (Y*) BIT 15 INTO CC; 1 = Y* BITS 15,14 05994000 D
                                                          % SET CC          05995000 D
125C  238C 0000 0000 0060          CONTENTSRM - 1 = CPCR  % Y* INTO B       05996000 D
125D  4809 0C41 0C10 00FC          B L = BR2              % Y* INTO BR2      05997000 D
```

```
125E  0000 0000 0000 0030          COMP 8 = SAR                                          05998000 D
125F  50E0 0000 0000 0060          EMULIN - 1 = CPCR      % (Y*) INTO B                  05999000 D
1260  4809 0C41 1830 00F0          B L = B, A3            % (Y*) INTO MS WORD OF B, A3    06000000 D
1261  0000 0000 0000 0020          COMP 16 = SAR                                         06001000 D
1262  4809 CC40 0000 00F0          B                      % P TO THE ADDER                06002000 D
1263  400E 0000 0000 00F0          IF NOT MST THEN STEP ELSE SKIP                        06003000 D
1264  1A00 0000 0000 0040          P351 - 1 = MPCR        % POSITIVE CASE                 06004000 D
1265  2160 0000 0000 0060          SET11 - 1 = CPCR       % SET THE CONDITION BITS        06005000 D
1266  4809 0001 0B30 00F0          1 L = B                                               06006000 D
1267  2000 0000 0000 0000          COMP 30 = SAR                                         06007000 D
1268  4809 EC50 8830 00F0          A3 OR B R = MIR, B     % SET BIT 14 OF (Y*)           06008000 D
1269  0000 0000 0000 0020          16 = SAR                                              06009000 D
126A  4809 0F41 0010 00F0          BMAR L = BR2           % RESTORE Y* INTO BR2          06010000 D
126B  0000 0000 0000 0030          COMP 8 = SAR                                          06011000 D
126C  51E0 0000 0000 0060          EMULOUT - 1 = CPCR                                    06012000 D
126D  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                     06013000 D
126E  4809 A001 0000 40F0  P351:   A1 C = A1,CSAR         % PUT CC BIT 9 INTO LS BIT     06014000 D
126F  1000 0000 0000 0030          25 = SAR                                              06015000 D
1270  4809 A0CF 0000 00F0          A1 AND B110 C = A1     % SET BIT 9 TO 0, AND RESTORE A1 06016000 D
1271  4809 0C41 8800 00F0          B C = B                                               06017000 D
1272  A000 0000 0000 0030          30 = SAR                                              06018000 D
1273  4809 2C50 8030 00F0          LIT OR B C = MIR       % SET (Y*) BITS 14,15           06019000 D
1274  2030 0000 0000 00A0          3 = LIT; 18 = SAR                                     06020000 D
1275  4809 0F41 0010 00F0          BMAR L = BR2           % RESTORE Y* INTO BR2          06021000 D
1276  0000 0000 0000 0030          COMP 8 = SAR                                          06022000 D
1277  51E0 0000 0000 0060          EMULOUT - 1 = CPCR                                    06023000 D
1278  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                     06024000 D
                                                         %                               06025000 D
                                                         %                               06026000 D
                           OP352:                        % TYPE RK REMOTE EXECUTE        06027000 D
                                                         % EXECUTE (Y); (P) + 2 = P      06028000 D
1279  4809 2C56 0B30 00F0          LIT AND P = B          % ISOLATE "M"                   06029000 D
127A  00F0 0000 0000 00E0          15 = LIT                                              06030000 D
127B  4809 CC52 0000 00F0          0 EQL B                                               06031000 D
127C  6819 0000 0000 00F0          IF TRUE THEN SKIP                                     06032000 D
127D  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                06033000 D
127E  4809 0C41 0B30 00F0          B L = B                                               06034000 D
127F  0000 0000 0000 0020          COMP 16 = SAR                                         06035000 D
1280  4809 EC50 1000 00F0          A3 OR B = A3           % A3 = (R(M)) OR 0 /INSTRUCTION 06036000 D
1281  6330 0000 0000 0060          IFETCH - 1 = CPCR      % "Y" INTO B                   06037000 D
1282  4809 E000 A000 00F0          A3 R = A2              % (R(M)) OR 0 INTO A2           06038000 D
1283  0200 0000 0000 00A0          PSW = LIT; 16 = SAR                                   06039000 D
1284  4809 CC40 0030 00F0          A2 + B = MIR           % Y INTO MIR                    06040000 D
1285  4809 2000 0010 00F0          LIT = MAR2                                            06041000 D
1286  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR     % Y INTO PSW                   06042000 D
1287  4809 B81C 4000 00F0          A1 OR B1C0 = A1        % SET BIT 31 OF SR01, REMOTE EXEC BIT 06043000 D
1288  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                     06044000 D
                                                         %                               06045000 D
                                                         %                               06046000 D
                           OP353:                        % TYPE RX BIASED FETCH, (Y) BIT 15 IN CC 06047000 D
                                                         % 1 = (Y) BIT 15                06048000 D
1289  5700 0000 0000 0060          RXMFIELD - 1 = CPCR     % Y INTO B                    06049000 D
128A  4809 0C41 1010 00F0          B L = BR2, A3                                         06050000 D
128B  0000 0000 0000 003C          COMP 8 = SAR                                          06051000 D
128C  60E0 0000 0000 0060          EMULIN - 1 = CPCR      % (Y) INTO B                    06052000 D
128D  4809 0C41 0B30 00F0          B L = B                                               06053000 D
128E  0000 0000 0000 0020          COMP 16 = SAR                                         06054000 D
128F  4809 CC40 0000 00F0          B                                                     06055000 D
1290  400B 0000 0000 00F0          IF NOT MST THEN STEP ELSE SKIP                        06056000 D
1291  1A00 0000 0000 0040          P353 - 1 = MPCR                                       06057000 D
```

```
1292  2160 0000 0000 0060          SET11 - 1 = CPCR       % SET THE CONDITION BITS          06058000 D
1293  4809 00C1 2000 00F0          1 L = A2                                                 06059000 D
1294  2000 0000 0C30 0000          COMP 30 = SAR                                            06060000 D
1295  4809 CC5C 8B3C C0F0          A2 OR 6 R = MIR, C     % SET BIT 14 OF (Y)               06061000 D
1296  0000 0000 0C0C 0020          16 = SAR                                                 06062000 D
1297  4809 E0C0 0010 C0FC          A3 = BR2               % RESTORE Y INTO BR2              06063000 D
1298  61E0 0000 0000 0060          EMULOUT - 1 = CPCR                                       06064000 D
1299  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                        06065000 D
129A  4809 A0C1 CC0C 40F0  P353:   A1 C = A1,CSAR         % PUT CC BIT 9 INTO LS BIT        06066000 D
129B  1000 0000 0000 003C          25 = SAR                                                 06067000 D
129C  4809 A0CF C000 00F6          A1 AND B110 C = A1     % SET BIT 9 TO 0, AND RESTORE A1  06068000 D
129D  4809 0C41 8B3C 00F0          B C = B                                                  06069000 D
129E  A000 0000 0030 0030          30 = SAR                                                 06070000 D
129F  4809 2C50 8C3C 00F0          LIT OR B C = MIR       % SET (Y) BIT 14,15               06071000 D
12A0  2030 0000 0000 00A0          3 = LIT, 18 = SAR                                        06072000 D
12A1  4809 E0C0 C010 00F0          A3 = BR2               % RESTORE Y INTO BR2              06073000 D
12A2  61E0 0000 0000 0060          EMULOUT - 1 = CPCR                                       06074000 D
12A3  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                        06075000 D
                                                          %                                 06076000 D
                                                          %                                 06077000 D
                           OPCODE36:                      % NOT ASSIGNED. GENERATES INTERRUPT 06078000 D
                                                          % FAULT.                          06079000 D
12A4  3000 0000 0030 0040          FAULT - 1 = MPCR       % JUMP TO FAULT HANDLING ROUTINE  06080000 D
                                                          %                                 06081000 D
                                                          %                                 06082000 D
                           OPCODE37:                      % NOT IMPLEMENTED                  06083000 D
12A5  4E50 0000 0030 0040          NOTIMP - 1 = MPCR                                        06084000 D
                                                          %                                 06085000 D
                           OPCODE40:                      % THIS ROUTINE ANALYZES THE 40 OPCODE 06086000 D
12A6  5F90 0000 0000 006C          XFCODE - 1 = CPCR      % :F: FIELD RETURNED IN A2        06087000 D
12A7  4809 C0D2 0C0C 00FC          A2 EQL 1                                                 06088000 D
12A8  6806 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP                              06089000 D
12A9  1AEC 0000 0030 0040          OP401 - 1 = MPCR       % SPECIAL CASE 401                06090000 D
12AA  4809 0C40 8B0C 0CFC          B R = B                % ISOLATE THE :A: FIELD            06091000 D
12AB  90FC 0000 0000 0080          4 = SAR, 15 = LIT                                        06092000 D
12AC  4809 2C56 2B30 00FC          LIT AND B = B,A2       % B AND A2 = :A: FIELD             06093000 D
12AD  4809 2C5E 0000 00F0          LIT GEQ R                                                06094000 D
12AE  0000 0000 0C0C 00E0          12 = LIT               % TEST FOR NOT ASSIGNED INST.      06095000 D
12AF  7819 0000 0030 00F0          IF TRUE THEN SKIP                                        06096000 D
12B0  3000 0000 0030 0040          FAULT - 1 = MPCR                                         06097000 D
12B1  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                       06098000 D
12B2  1AFC 0000 0000 C0C0          JUMP40 - 1 = AMPCR                                       06099000 D
12B3  4809 0000 0030 00F0          STEP                                                     06100000 D
12B4  4824 0000 0030 00F0          EXEC                                                     06101000 D
                                                          %                                 06102000 D
12B5  1B00 0000 0030 0040  JUMP40: OP40X00 - 1 = MPCR                                       06103000 D
12B6  1B10 0000 0030 0040          OP40X01 - 1 = MPCR                                       06104000 D
12B7  1B20 0000 0000 C040          OP40X02 - 1 = MPCR                                       06105000 D
12B8  1B30 0000 0000 0040          OP40X03 - 1 = MPCR                                       06106000 D
12B9  1B40 0000 0000 C040          OP40X04 - 1 = MPCR                                       06107000 D
12BA  1B50 0000 0000 004C          OP40X05 - 1 = MPCR                                       06108000 D
12BB  4E50 0000 0030 0040          NOTIMP - 1 = MPCR                                        06109000 D
12BC  4E50 0000 0030 C040          NOTIMP - 1 = MPCR                                        06110000 D
12BD  1B60 0000 0000 0040          OP40X10 - 1 = MPCR                                       06111000 D
12BE  1B70 0000 0000 C040          OP40X11 - 1 = MPCR                                       06112000 D
12BF  1B80 0000 0000 C040          OP40X12 - 1 = MPCR                                       06113000 D
12C0  1B90 0000 0000 C040          OP40X13 - 1 = MPCR                                       06114000 D
                                                          %                                 06115000 D
                           OP40X00:                       % CC INDICATES = OR 0; JUMP EQUAL  06116000 D
12C1  226C 0000 0000 0060          CHECKCC - 1 = CPCR     % CC RETURNED IN B                06117000 D
```

```
12C2  4809 0C52 0000 00F0          0 EQL B                                   06118000 D
12C3  63C9 0000 0000 00F0          IF FALSE THEN SET LC1                     06119000 D
12C4  18A0 0000 000C 0040          OP40X - 1 = MPCR                          06120000 D
                                                                             06121000 D
                                   %                                         06121000 D
OP40X01:                           % CC NOT EQUAL OR NOT 0; JUMP NOT EQUAL   06122000 D
12C5  2260 00C0 0000 C060          CHECKCC - 1 = CPCR                        06123000 D
12C6  4809 0C5E 0000 00F0          0 GTR 0                                   06124000 D
12C7  78C9 0000 0030 00F0          IF FALSE THEN SET LC1                     06125000 D
12C8  18B0 0000 0030 0040          OP40X - 1 = MPCR                          06126000 D
                                                                             06127000 D
                                   %                                         06127000 D
OP40X02:                           % CC >= OR + ; JUMP GTR OR EQUAL          06128000 D
12C9  2260 00C0 0000 0060          CHECKCC - 1 = CPCR                        06129000 D
12CA  4809 2C5E 0030 00F0          LIT GEQ B                                 06130000 D
12CB  0020 0000 0000 00E0          2 = LIT                                   06131000 D
12CC  73C9 0000 0000 00F0          IF FALSE THEN SET LC1                     06132000 D
12CD  18C0 0000 000C 0040          OP40X - 1 = MPCR                          06133000 D
                                                                             06134000 D
                                   %                                         06134000 D
OP40X03:                           % CC < OR -; JUMP LESS                    06135000 D
12CE  2260 0000 0030 0060          CHECKCC - 1 = CPCR                        06136000 D
12CF  4809 2C52 0030 00F0          LIT EQL E                                 06137000 D
12D0  0030 0000 0030 00E0          3 = LIT                                   06138000 D
12D1  53C9 0000 0000 00F0          IF FALSE THEN SET LC1                     06139000 D
12D2  18D0 0000 000C 0040          OP40X - 1 = MPCR                          06140000 D
                                                                             06141000 D
                                   %                                         06141000 D
OP40X04:                           % IF OVERFLOW SET, JUMP; JUMP OVERFLOW    06142000 D
12D3  4809 A000 A000 00F0          A1 R = A2                                 06143000 D
12D4  3000 0000 0000 0030          27 = SAR                                  06144000 D
12D5  4809 0000 0030 00F0          A2                                        06145000 D
12D6  53C9 0000 0000 00F0          IF NOT LST THEN SET LC1                   06146000 D
12D7  18E0 0000 000C 0040          OP40X - 1 = MPCR                          06147000 D
                                                                             06148000 D
                                   %                                         06148000 D
OP40X05:                           % IF CARRY; JUMP CARRY                    06149000 D
12D8  4809 A000 A030 00F0          A1 R = A2                                 06150000 D
12D9  8000 0000 0030 0030          28 = SAR                                  06151000 D
12DA  4809 0000 0030 00F0          A2                                        06152000 D
12DB  53C9 0000 0030 00F0          IF NOT LST THEN SET LC1                   06153000 D
12DC  18F0 0000 0000 0040          OP40X - 1 = MPCR                          06154000 D
                                                                             06155000 D
                                   %                                         06155000 D
OP40X10:                           % JUMP                                    06156000 D
12DD  1000 0000 0000 0040          OP40X - 1 = MPCR                          06157000 D
                                                                             06158000 D
                                   %                                         06158000 D
OP40X11:                           % JUMP AFTER STOP                         06159000 D
12DE  4800 0000 0000 00F0          WAIT                  % MACHINE STOPS     06160000 D
12DF  4809 0000 000C 00F0          STEP                                      06161000 D
12E0  1010 0000 000C 0040          OP40X - 1 = MPCR                          06162000 D
                                                                             06163000 D
                                   %                                         06163000 D
OP40X12:                           % IF KEY 1 SET, STOP. JUMP.               06164000 D
12E1  4809 A000 A030 00F0          A1 R = A2                                 06165000 D
12E2  9000 0000 0000 0030          29 = SAR                                  06166000 D
12E3  4809 0000 0030 00F0          A2                                        06167000 D
12E4  53D0 0000 0000 00F0          IF NOT LST THEN SET LC1 ELSE SKIP         06168000 D
12E5  1020 0000 0000 0040          OP40X - 1 = MPCR                          06169000 D
12E6  4800 0000 0000 00F0          WAIT                  % MACHINE STOPS     06170000 D
12E7  4809 0000 000C 00F0          STEP                                      06171000 D
12E8  1030 0000 000C 0040          OP40X - 1 = MPCR                          06172000 D
                                                                             06173000 D
                                   %                                         06173000 D
OP40X13:                           % IF KEY 2 SET, STOP. JUMP                06174000 D
12E9  4809 A000 A000 00F0          A1 R = A2                                 06175000 D
12EA  A000 0000 0000 0030          30 = SAR                                  06176000 D
12EB  4809 0000 0030 00F0          A2                                        06177000 D
```

```
12EC  53CB 0000 0000 00F0          IF NOT LST THEN SET LC1 ELSE SKIP                 06178000 D
12ED  5819 0000 0000 C0F0          IF LST THEN SKIP                                  06179000 D
12EE  1C40 0000 0000 0040          OP40X - 1 = MPCR                                  06180000 D
12EF  4800 0000 0000 00F0          WAIT                      % MACHINE STOPS         06181000 D
12F0  4809 0000 0000 00F0          STEP                                              06182000 D
12F1  1C50 0000 0000 0040          OP40X - 1 = MPCR                                  06183000 D
                                                             %                       06184000 D
12F2  5F90 0000 0000 0060  OP40X:  XFCODE - 1 = CPCR         % DECODE THE :F: FIELD   06185000 D
12F3  4809 C640 0000 00FC          A2 + AMPCR = AMPCR                                06186000 D
12F4  1E60 0000 0000 00C0          OP40F - 1 = AMPCR                                 06187000 D
12F5  4809 0000 0000 00F0          STEP                                              06188000 D
12F6  4824 0000 0000 00F0          EXEC                                              06189000 D
                                                             %                       06190000 D
12F7  1C70 0000 0000 0040  OP40F:  OP400 - 1 = MPCR                                  06191000 D
12F8  3000 0000 0000 0040          FAULT - 1 = MPCR          % DECODING PROBLEM      06192000 D
12F9  1C80 0000 0000 0040          OP402 - 1 = MPCR                                  06193000 D
12FA  1C90 0000 0000 0040          OP403 - 1 = MPCR                                  06194000 D
                                                             %                       06195000 D
                           OP400:                            % THIS ROUTINE LOADS (R(M)) INTO P.  06196000 D
                                                             % TYPE RR                06197000 D
12FB  280B 0000 0000 00F0          IF LC1 THEN STEP ELSE SKIP  % ABORT JUMP, GET NEXT INST.  06198000 D
12FC  66E0 0000 0000 004C          OPCODE - 1 = MPCR                                 06199000 D
12FD  4809 E155 0B00 00FC          A3 AND LIT = B            % B = :M: FIELD         06200000 D
12FE  00FC 0000 0000 C0EC          15 = LIT                                          06201000 D
12FF  51CC 0000 0000 0060          REGSTACK - 1 = CPCR                               06202000 D
1300  2F30 0000 0000 006C          EINPUT - 1 = CPCR         % B = (R(M))            06203000 D
1301  4809 A003 0000 00FC          A1 R = A1                                         06204000 D
1302  0000 0000 0000 0020          16 = SAR                                          06205000 D
1303  4809 A001 4000 00F0          A1 L = A1                                         06206000 D
1304  4809 AC5C 4000 00F0          A1 OR B = A1              % (R(M)) -> P            06207000 D
1305  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                 06208000 D
                                                             %                       06209000 D
                           OP402:                            % THIS ROUTINE LOADS Y INTO P. TYPE RK  06210000 D
1306  280B 0000 0000 00F0          IF LC1 THEN STEP ELSE SKIP  % ABORT JUMP, GET NEXT INST.  06211000 D
1307  1D30 0000 0000 0040          BUMP - 1 = MPCR                                   06212000 D
1308  4809 E155 0B00 00F0          A3 AND LIT = B                                    06213000 D
1309  00FC 0000 0000 C0E0          15 = LIT                                          06214000 D
130A  4809 0C52 0000 00F0          0 EQL B                                           06215000 D
130B  6A49 0000 0000 00FC          IF TRUE THEN SET LC2                              06216000 D
130C  51CC 0000 0000 0060          REGSTACK - 1 = CPCR                               06217000 D
130D  2F30 0000 0000 0060          EINPUT - 1 = CPCR         % B = (R(M))            06218000 D
130E  4809 0C40 1000 00F0          B = A3                    % A3 = (R(M))            06219000 D
130F  633C 0000 0000 0060          IFETCH - 1 = CPCR         % B = :Y:                06220000 D
1310  3819 0000 0000 00FC          IF LC2 THEN SKIP                                  06221000 D
1311  280B 0000 0000 00F0          IF LC1 THEN STEP ELSE SKIP  % ABORT JUMP, GET NEXT INST.  06222000 D
1312  1D30 0000 0000 0040          BUMP - 1 = MPCR                                   06223000 D
1313  4809 EC40 0B00 00FC          A3 + B = B                % (R(M)) + :Y: = Y       06224000 D
1314  4809 A003 0000 00F0          A1 R = A1                                         06225000 D
1315  0000 0000 0000 0020          16 = SAR                                          06226000 D
1316  4809 A001 4000 00F0          A1 L = A1                                         06227000 D
1317  4809 AC5C 4000 00F0          A1 OR B = A1              % Y INTO PAR            06228000 D
1318  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                 06229000 D
                                                             %                       06230000 D
                           OP403:                            % THIS ROUTINE LOADS Y INTO P.  06231000 D
                                                             % TYPE RX.              06232000 D
1319  5700 0000 0000 0060          RXMFIELD - 1 = CPCR       % B = Y                  06233000 D
131A  4809 C041 0010 00F0          B L = BR2                                         06234000 D
131B  0000 0000 0000 0030          COMP B = SAR                                      06235000 D
131C  50E0 0000 0000 0060          EMULIN - 1 = CPCR         % B = (Y)                06236000 D
131D  4809 A000 0000 00F0          A1 R = A1                                         06237000 D
```

```
131E  0000 0000 0000 0020          16 = SAR                                  06238000 D
131F  4809 ACC1 4000 00F0          A1 L = A1                                 06239000 D
1320  4809 AC5C 4000 00F0          A1 OR B = A1          % (Y) INTO PAR       06240000 D
1321  66E0 0000 0000 0040          OPCODE - 1 = MPCR                         06241000 D
                                                         %                    06242000 D
                                OP401:                   % LJ: LOCAL JUMP      06243000 D
                                                         % (P) + XD -> P      06244000 D
                                                         % TYPE RI (1)        06245000 D
1322  559C 0000 0000 004C          RI1 - 1 = MPCR        % (P) + XD -> P      06246000 D
                                                         %                    06247000 D
                                                         %                    06248000 D
                                OPCODE41:                % INDEX JUMP          06249000 D
1323  5F90 0000 0000 0060          XFCODE - 1 = CPCR     % "F" CODE RETURNED IN A2  06250000 D
1324  4809 C640 004C 00FC          A2 + AMPCR = AMPCR                        06251000 D
1325  1CAC 0003 0000 00C0          OP41F - 1 = AMPCR                         06252000 D
1326  4809 0000 0000 00F0          STEP                                      06253000 D
1327  4824 0003 0000 00FC          EXEC                                      06254000 D
                                                         %                    06255000 D
                                                         %                    06256000 D
1328  1C80 0000 0000 0040  OP41F:  OP410 - 1 = MPCR                          06257000 D
1329  1CCC 0000 0000 0040          OP411 - 1 = MPCR                          06258000 D
132A  1C00 0000 0000 0040          OP412 - 1 = MPCR                          06259000 D
132B  1CEC 0000 0000 0040          OP413 - 1 = MPCR                          06260000 D
                                                         %                    06261000 D
                                                         %                    06262000 D
                                OP410:                   % RR TYPE INDEX JUMP  06263000 D
                                                         % IF (R(A)) NE 0, (R(A)) - 1 INTO R(A)  06264000 D
                                                         % AND (R(M)) IN P     06265000 D
132C  2260 0000 0000 0060          CONTENTSRA - 1 = CPCR  % RETURN (R(A)) IN B  06266000 D
132D  4809 0052 0000 00F0          B EQL 0                                   06267000 D
132E  680B 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP               06268000 D
132F  66E0 0000 0000 0040          OPCODE - 1 = MPCR                         06269000 D
1330  4809 0040 200C 00F0          B = A2                                    06270000 D
1331  4809 C0DE 003C 00F0          A2 - 1 = MIR          % DECREMENT (R(A))   06271000 D
1332  2F50 0000 0000 0060          EOUTPUT - 1  = CPCR   % WRITE NEW (R(A)) VALUE IN R(A)  06272000 D
1333  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % B CONTAINS (R(M))  06273000 D
1334  4809 ACC0 C000 00F0          A1 R = A1             % CLEAR THE OLD PAR   06274000 D
1335  0000 0000 0000 0020          16 = SAR                                  06275000 D
1336  4809 ACC1 4000 00F0          A1 L = A1                                 06276000 D
1337  4809 AC5C 4000 00F0          A1 OR B = A1          % CREATE NEW PAR      06277000 D
1338  66E0 0000 0000 0040          OPCODE - 1 = MPCR                         06278000 D
                                                         %                    06279000 D
                                                         %                    06280000 D
                                OP411:                   % RI TYPE 1, LOCAL JUMP INDIRECT  06281000 D
                                                         % IF (CC) NE 0, ((P) + D) INTO P  06282000 D
1339  2260 0000 0000 0060          CHECKCC - 1 = CPCR    % PUT (CC) INTO LS 2 BITS OF B  06283000 D
133A  4809 0052 0000 00F0          B EQL 0                                   06284000 D
133B  680B 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP               06285000 D
133C  66E0 0000 0000 0040          OPCODE - 1 = MPCR                         06286000 D
133D  5590 0000 0000 004C          RI1 - 1 = MPCR        % (P) + D = P        06287000 D
                                                         %                    06288000 D
                                                         %                    06289000 D
                                OP412:                   % PK TYPE INDEX JUMP  06290000 D
                                                         % IF (R(A)) NE 0, (R(A)) - 1 INTO R(A),  06291000 D
                                                         % Y INTO P            06292000 D
133E  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR  % PUT (R(A)) IN B    06293000 D
133F  4809 0052 0000 00F0          B EQL 0                                   06294000 D
1340  680B 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP               06295000 D
1341  66E0 0000 0000 0040          OPCODE - 1 = MPCR                         06296000 D
1342  4809 0040 200C 00F0          B = A2                                    06297000 D
```

```
1343  4809 C0DE C090 C0F0          A2 - 1 = MIR          % DECREMENT (R(A))                  06298C00 D
1344  4809 E0C3 80LC 00F0          A3 R = MAR2           % ADDRESS OF R(A) INTO MAR2         06299C00 D
1345  0000 0000 0030 0020          16 = SAR                                                   06300C00 D
1346  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR    % (R(A)) - 1  INTO R(A)             06301C00 D
1347  4809 E155 100C C0F0          A3 AND LIT = A3       % "M" FIELD IN A3                   06302C00 D
1348  00FC 0000 0000 00EC          15 = LIT                                                   06303C00 D
1349  4809 E012 0030 00F0          A3 EQL C                                                   06304C00 D
134A  6C19 0000 0000 00F0          IF TRUE THEN 0 = 0; SKIP                                   06305C00 D
134B  238C 00C0 0030 006C          CONTENTSRM - 1 = CPCR  % RETURN (R(M)) IN B                06306C00 D
134C  4809 0C40 1000 00FC          B = A3                % STORE (R(M)) OR C IN A3           06307C00 D
134D  633C 0000 0000 0060          IFETCH - 1 = CPCR     % PUT Y IN B                       06308C00 D
134E  4809 EC40 C800 00F0          A3 + B = B            % Y INTO B                          06309C00 D
134F  4809 A000 C000 C0F0          A1 R = A1                                                  06310C00 D
1350  0000 0000 0030 0020          COMP 16 = SAR                                              06311C00 D
1351  4809 A001 4030 00F0          A1 L = A1                                                  06312C00 D
1352  4809 AC5C 4000 00F0          A1 OR 0 = A1                                               06313C00 D
1353  56EC 0000 0000 004C          OPCODE - 1 = MPCR                                          06314C00 D
                                                         %                                    06315C00 D
                                                         %                                    06316C00 D
                                   OP413:                % RX TYPE INDEX JUMP                 06317C00 D
                                                         % IF (R(A)) NE 0, (R(A)) - 1 = (R(A)), 06318C00 D
                                                         % (Y) INTO P                         06319C00 D
1354  4809 E000 8830 00FC          A3 R = B              % SHIFT A3 TO OBTAIN "A"            06320C00 D
1355  80F0 00C0 0000 008C          4 = SAR; 15 = LIT                                          06321C00 D
1356  4809 2C56 0800 00F0          LIT AND B = B         % ISOLATE "A"                       06322C00 D
1357  51C0 0000 0030 0060          REGSTACK - 1 = CPCR   % R(A) INTO MAR2                    06323C00 D
1358  2F30 00C0 0030 C060          EINPUT - 1 = CPCR     % (R(A)) INTO B                    06324C00 D
1359  4809 0C52 0030 C0F0          B EQL C                                                    06325C00 D
135A  580F 0000 0030 C0F0          IF TRUE THEN STEP ELSE SKIP                                06326C00 D
135B  1030 00C0 0000 0040          BUMP - 1 = MPCR       % ADVANCE THE PAR BY 1 AND CALL OPCODE 06327C00 D
135C  4809 0C40 2000 00FC          B = A2                                                     06328C00 D
135D  4809 C0DE C090 C0FC          A2 - 1 = MIR          % DECREMENT (R(A))                  06329C00 D
135E  2F50 0000 0000 C060          EOUTPUT - 1 = CPCR    % OUTPUT NEW VALUE OF R(A) IN R(A)  06330C00 D
135F  4809 E000 0800 00F0          A3 = B                % RESTORE INSTRUCTION IN B          06331C00 D
1360  5700 00C0 0030 0060          RXMFIELD - 1 = CPCR   % ANALYZE THE "M" FIELD             06332C00 D
1361  4809 0C41 0010 00FC          B L = BR2             % ADDRESS OF Y IN BR2               06333C00 D
1362  0000 0000 0030 0030          COMP 8 = SAR                                               06334C00 D
1363  60EC 00C0 0000 C060          EMULIN - 1 = CPCF     % (Y) INTO B                       06335C00 D
1364  4809 A000 C000 00F0          A1 R = A1             % CLEAR OLD PAR                     06336C00 D
1365  0000 0000 0000 0020          COMP 16 = SAR                                              06337C00 D
1366  4809 A001 4030 C0F0          A1 L = A1                                                  06338C00 D
1367  4809 AC5C 4030 00FC          A1 OR B = A1          % CREATE NEW PAR                    06339C00 D
1368  66E0 0000 0030 C040          OPCODE - 1 = MPCR                                          06340C00 D
                                                         %                                    06341C00 D
                                                         %                                    06342C00 D
                                   OPCODE42:             % RR TYPE JUMP, LINK REGISTERS      06343C00 D
1369  5F90 00C0 0030 C060          XFCODE - 1 = CPCR     % "F" INTO A2                      06344C00 D
136A  4809 C640 0040 00FC          A2 + AMPCR = AMPCR                                         06345C00 D
136B  1CF0 00C0 0000 00C0          OP42F - 1 = AMPCR                                          06346C00 D
136C  4809 00C0 0030 00FC          STEP                                                       06347C00 D
136D  4824 00C0 0000 00F0          EXEC                                                       06348C00 D
                                                         %                                    06349C00 D
                                                         %                                    06350C00 D
136E  1000 0000 0000 004C          OP42F:  OP420 - 1 = MPCR                                   06351C00 D
136F  3000 0000 0030 C040          FAULT - 1 = MPCR                                           06352C00 D
1370  1010 0000 0000 C04C          OP422 - 1 = MPCR                                           06353C00 D
1371  1D20 00C0 0030 C040          OP423 - 1 = MPCR                                           06354C00 D
                                                         %                                    06355C00 D
                                                         %                                    06356C00 D
                                   OP420:                % JUMP, LINK REGISTER               06357C00 D
```

```
                                                        % (P) + 1 INTO R(A); (R(M)) INTO P      06358000 D
1372  4809 0C40 8800 00F0          B R = B                                                       06359000 D
1373  30F0 0000 0000 0080          4 = SAR; 15 = LIT                                             06360000 D
1374  4809 2056 0800 00F0          LIT AND B = B          % B CONTAINS "A"                       06361000 D
1375  5100 0000 0000 0060          REGSTACK - 1 = CPCR    % ADDRESS OF R(A) IN MAR2              06362000 D
1376  4809 A001 2000 00F0          A1 L = A2              % (P) INTO MS WORD OF A2               06363000 D
1377  0000 0000 0000 0020          16 = SAR                                                      06364000 D
1378  4809 C000 A000 00F0          A2 R = A2              % (P) INTO LS WORD OF A2               06365000 D
1379  4809 C000 0030 00F0          A2 + 1 = MIR           % (P) + 1 INTO MIR                    06366000 D
137A  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR     % (P) + 1 INTO R(A)                   06367000 D
137B  2380 0000 0000 006C          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                       06368000 D
137C  4809 A000 C000 00F0          A1 R = A1              % CLEAR PAR FIELD                      06369000 D
137D  0000 0000 0000 0020          COMP 16 = SAR                                                 06370000 D
137E  4809 A001 4000 00F0          A1 L = A1              % PREPARE FOR NEW PAR                  06371000 D
137F  4809 AC50 4000 00F0          A1 OR B = A1           % CREATE NEW PAR                       06372000 D
1380  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                             06373000 D
                                                        %                                        06374000 D
                                                        %                                        06375000 D
                                OP422:                  % RK TYPE JUMP, LINK REGISTERS            06376000 D
                                                        % (P) + 2 INTO R(A); Y INTO P             06377000 D
1381  4809 0C40 8800 00F0          B R = B                                                       06378000 D
1382  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                             06379000 D
1383  4809 2056 0800 00F0          LIT AND B = B          % B CONTAINS "A"                       06380000 D
1384  5100 0000 0000 0060          REGSTACK - 1 = CPCR    % MAR2 CONTAINS ADDRESS OF R(A)        06381000 D
1385  4809 A001 2000 00F0          A1 L = A2              % SAVE OLD PAR                         06382000 D
1386  0020 0000 0000 00A0          COMP 16 = SAR; 2 = LIT                                        06383000 D
1387  4809 C000 A000 00F0          A2 R = A2              % PAR INTO LS WORD OF A2               06384000 D
1388  4809 C140 0030 00F0          A2 + LIT = MIR         % MIR CONTAINS (P) + 2                 06385000 D
1389  2F50 0000 0000 006C          EOUTPUT - 1 = CPCR     % PUT (P) + 2 INTO R(A)                06386000 D
138A  4809 E156 1000 00F0          A3 AND LIT = A3        % "M" INTO A3                          06387000 D
138B  00F0 0000 0000 00E0          15 = LIT                                                      06388000 D
138C  4809 E012 0000 00F0          A3 EQL 0                                                      06389000 D
138D  6C19 0000 0000 00FC          IF TRUE THEN 0 = 0; SKIP  % CHECK IF "M" = 0                  06390000 D
138E  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                       06391000 D
138F  4809 0C40 1000 00F0          B = A3                                                        06392000 D
1390  633C 0000 0000 0060          IFETCH - 1 = CPCR      % Y INTO B                            06393000 D
1391  4809 EC40 C800 00F0          A3 + B = B             % (Y) INTO B                           06394000 D
1392  4809 A000 C000 00F0          A1 R = A1              % CLEAR THE PAR                        06395000 D
1393  0000 0000 0000 0020          COMP 16 = SAR                                                 06396000 D
1394  4809 A001 4000 00F0          A1 L = A1              % PREPARE FOR THE NEW PAR              06397000 D
1395  4809 AC50 4000 00F0          A1 OR B = A1           % CREATE THE NEW PAR                   06398000 D
1396  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                             06399000 D
                                                        %                                        06400000 D
                                                        %                                        06401000 D
                                OP423:                  % RX TYPE JUMP, LINK REGISTERS            06402000 D
                                                        % (P) + 2 INTO R(A); (Y) INTO P           06403000 D
1397  4809 0C40 8800 00F0          B R = B                                                       06404000 D
1398  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                             06405000 D
1399  4809 2056 0800 00F0          LIT AND B = B          % B CONTAINS "A"                       06406000 D
139A  5100 0000 0000 0060          REGSTACK - 1 = CPCR    % ADDRESS OF R(A) IN MAR2              06407000 D
139B  4809 A001 2000 00F0          A1 L = A2              % (P) INTO MS WORD OF A2               06408000 D
139C  0020 0000 0000 00A0          16 = SAR; 2 = LIT                                             06409000 D
139D  4809 C000 A000 00F0          A2 R = A2              % (P) INTO LS WORD OF A2               06410000 D
139E  4809 C140 0030 00F0          A2 + LIT = MIR         % (P) + 2 INTO MIR                    06411000 D
139F  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR     % (P) + 2 INTO R(A)                   06412000 D
13A0  4809 E000 0800 00F0          A3 = B                 % SET UP B FOR RXMFIELD                06413000 D
13A1  57C0 0000 0000 0060          RXMFIELD - 1 = CPCR    % Y INTO B                            06414000 D
13A2  4809 0C41 0C10 00F0          B L = BR2              % Y INTO BR2                           06415000 D
13A3  00C0 0000 0000 0030          COMP 8 = SAR                                                  06416000 D
13A4  60E0 0000 0000 0060          EMULIN - 1 = CPCR      % (Y) INTO B                          06417000 D
```

```
13A5  4809 A000 0000 00F0          A1 R = A1                  % CLEAR THE OLD PAR          06418000 D
13A6  0000 0000 0000 0020          COMP 16 = SAR                                           06419000 D
13A7  4809 A001 4000 00F0          A1 L = A1                  % PREPARE TO RECEIVE NEW PAR  06420000 D
13A8  4809 A05C 4000 00F0          A1 OR B = A1               % CREATE NEW PAR              06421000 D
13A9  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                       06422000 D
                                                              %                            06423000 D
                                                              %                            06424000 D
                                   OPCODE43:                  % JUMP, LINK MEMORY           06425000 D
13AA  5F90 0000 0000 0060          XFCODE - 1 = CPCR          % "F" RETURNED IN A2          06426000 D
13AB  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                      06427000 D
13AC  103C 0000 0000 00C0          OP43F - 1 = AMPCR                                       06428000 D
13AD  4809 0000 0030 00F0          STEP                                                    06429000 D
13AE  4824 0000 0000 00F0          EXEC                                                    06430000 D
                                                              %                            06431000 D
                                                              %                            06432000 D
13AF  3000 0000 0000 0040          OP43F:  FAULT - 1 = MPCR                                06433000 D
1300  1040 0000 0000 0040          OP431 - 1 = MPCR                                        06434000 D
1301  1050 0000 0000 0040          OP432 - 1 = MPCR                                        06435000 D
1302  1060 0000 0000 0040          OP433 - 1 = MPCR                                        06436000 D
                                                              %                            06437000 D
                                                              %                            06438000 D
                                   OP431:                     % RI TYPE 1, LOCAL JUMP, LINK MEMORY  06439000 D
                                                              % ONLY EXECUTE IF CC = 0     06440000 D
1303  2260 0000 0000 0060          CHECKCC - 1 = CPCR         % CHECK THE CC CODE           06441000 D
1304  4809 0052 0000 00F0          0 EQL B                    % CHECK IF CC = 0             06442000 D
1305  6819 0000 0000 00F0          IF TRUE THEN SKIP                                       06443000 D
1306  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                       06444000 D
1307  2809 A001 2000 00F0          A1 L = A2; IF LC1          % PAR INTO A2 (MS WORD)       06445000 D
1308  0000 0000 0000 0020          COMP 16 = SAR                                           06446000 D
1309  4809 C000 A000 00F0          A2 R = A2                  % PAR INTO A2 (LS WORD)       06447000 D
130A  4809 C000 0030 00F0          A2 + 1 = MIR               % (P) + 1 INTO MIR            06448000 D
130B  4809 C001 2000 00F0          A2 L = A2                  % PAR INTO UHW                06449000 D
130C  4809 E000 0800 00F0          A3 = B                     % RESTORE INSTRUCTION INTO B  06450000 D
130D  4809 0041 0070 40F0          B L = B,CSAR               % "D" SIGN BIT IN MS BIT OF B 06451000 D
130E  0000 0000 0000 0010          COMP 24 = SAR                                           06452000 D
130F  4809 0040 0000 00F0          B                                                       06453000 D
1310  4FC9 0050 0000 00F0          IF MST THEN - B = B; SET LC1  % IF "D" NEGATIVE SET LC1  06454000 D
1311  4809 0040 8800 00F0          B R = B                    % "D" MAGNITUDE IN LSB OF B   06455000 D
1312  4809 2057 2830 00F0          LIT AND B L = B            % "D" MAGNITUDE IN B          06456000 D
1313  07F0 0000 0000 00A0          127 = LIT; COMP 16 = SAR                                06457000 D
1314  4809 0000 0030 00F0          STEP                                                    06458000 D
1315  0000 0000 0000 0010          B = SAR                                                 06459000 D
1316  2019 0050 9010 00F0          IF LC1 THEN A2 - B R = BR2,A3; SKIP  % (P) - D          06460000 D
1317  4809 0040 9010 00F0          A2 + B R = BR2,A3          % (P) + D                    06461000 D
1318  61E0 0000 0000 0060          CMULOUT - 1 = CPCR         % (P) + 1 INTO (P) + D       06462000 D
1319  4809 E000 8800 00F0          A3 R = B                                                06463000 D
131A  0000 0000 0030 0010          B = SAR                                                 06464000 D
131B  4809 0046 0830 00F0          B + 1 = B                  % (P) + D + 1 INTO B          06465000 D
131C  4809 A000 0000 00F0          A1 R = A1                  % CLEAR OLD PAR               06466000 D
131D  0000 0000 0000 0020          COMP 16 = SAR                                           06467000 D
131E  4809 A001 4000 00F0          A1 L = A1                                               06468000 D
131F  4809 A05C 4000 00F0          A1 OR B = A1               % CREATE NEW PAR              06469000 D
1300  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                       06470000 D
                                                              %                            06471000 D
                                                              %                            06472000 D
                                   OP432:                     % RK TYPE JUMP, LINK MEMORY   06473000 D
                                                              % (P) + 2 INTO Y; Y + 1 INTO P  06474000 D
1301  4809 2055 0800 00F0          LIT AND B = B              % ISOLATE "H" FIELD           06475000 D
1302  00F0 0000 0000 00E0          15 = LIT                                                06476000 D
1303  4809 0052 0000 00F0          B EQL 0                    % IS "H" = 0 ?                06477000 D
```

```
1304  6819 0000 0000 00F0          IF TRUE THEN SKIP                                  06478000 D
1305  2380 0000 0030 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B             06479000 D
1306  4809 0C40 1000 00F0          B = A3                                             06480000 D
1307  533C 0000 0000 0060          IFETCH - 1 = CPCR      % Y INTO B                  06481000 D
1308  4809 EC41 1010 00F0          A3 + B L = BR2,A3      % Y INTO BR2, A3             06482000 D
1309  0000 0000 0000 0030          COMP B = SAR                                       06483000 D
13DA  4809 A001 2030 00FC          A1 L = A2              % (P) INTO A2                06484000 D
13DB  002C 0000 0030 00A0          COMP 16 = SAR; 2 = LIT                             06485000 D
13DC  4809 C000 A000 00F0          A2 R = A2              % (P) INTO LS WORD OF A2    06486000 D
13DD  4809 C140 0030 00FC          A2 + LIT = MIR         % MIR CONTAINS (P) + 2      06487000 D
13DE  61E0 0000 0030 0060          EMULOUT - 1 = CPCR     % (P) + 2 INTO Y            06488000 D
13DF  4809 E000 9000 00F0          A3 R = A3              % Y INTO A3 LS WORD          06489000 D
13E0  0000 0000 0000 0010          B = SAR                                            06490000 D
13E1  4809 E000 0B30 00F0          A3 + 1 = B             % B CONTAINS Y + 1           06491000 D
13E2  4809 A003 C000 00F0          A1 R = A1              % CLEAR THE PAR              06492000 D
13E3  0000 0000 0000 0020          16 = SAR                                           06493000 D
13E4  4809 A001 4000 00F0          A1 L = A1                                          06494000 D
13E5  4809 AC3C 4000 00F0          A1 OR B = A1           % CREATE NEW PAR             06495000 D
13E6  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                  06496000 D
                                                          %                           06497000 D
                                                          %                           06498000 D
                                   OP433:                 % RX TYPE JUMP, LINK MEMORY  06499000 D
                                                          % (P) + 2 INTO (Y); (Y) + 1 INTO P  06500000 D
                                                          % B HOLDS THE INSTRUCTION    06501000 D
13E7  570C 0000 0000 0060          RXMFIELD - 1 = CPCR     % (Y) IN B                 06502000 D
13E8  4809 0C41 1010 00F0          B L = BR2, A3                                      06503000 D
13E9  000C 0000 0030 0030          COMP B = SAR                                       06504000 D
13EA  4809 A001 2030 00F0          A1 L = A2              % PUT PAR INTO A2            06505000 D
13EB  002C 0000 0030 00A0          COMP 16 = SAR; 2 = LIT                             06506000 D
13EC  4809 C000 A000 00F0          A2 R = A2              % (P) INTO A2               06507000 D
13ED  4809 C140 0030 00F0          A2 + LIT = MIR         % MIR CONTAINS (P) + 2      06508000 D
13EE  61E0 0000 0000 0060          EMULOUT - 1 = CPCR     % (P) + 2 INTO (Y)          06509000 D
13EF  4809 E000 9000 00F0          A3 R = A3              % (Y) INTO A3                06510000 D
13F0  000C 0000 0000 0010          B = SAR                                            06511000 D
13F1  4809 E000 0B30 00F0          A3 + 1 = B             % (Y) + 1 INTO B             06512000 D
13F2  4809 A000 C030 00F0          A1 R = A1                                          06513000 D
13F3  0000 0000 0000 0020          16 = SAR                                           06514000 D
13F4  4809 A001 4000 00F0          A1 L = A1              % A1 READY TO ACCEPT NEW PAR 06515000 D
13F5  4809 AC3C 4000 00F0          A1 OR B = A1           % CREATE NEW PAR             06516000 D
13F6  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                  06517000 D
                                                          %                           06518000 D
                                                          %                           06519000 D
                                   OPCODE44:              % JUMP ZERO                  06520000 D
13F7  5F9C 0000 0000 0060          XFCODE - 1 = CPCR      % RETURNS "F" CODE IN A      06521000 D
13F8  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                 06522000 D
13F9  1D70 0000 0030 00C0          OP44F - 1 = AMPCR                                  06523000 D
13FA  4809 0003 0030 00F0          STEP                                               06524000 D
13FB  4824 0000 0000 00F0          EXEC                                               06525000 D
                                                          %                           06526000 D
                                                          %                           06527000 D
13FC  1D80 0000 0030 0040          OP44F:  OP440 - 1 = MPCR                           06528000 D
13FD  1090 0000 0000 0040                  OP441 - 1 = MPCR                           06529000 D
13FE  10A0 0000 0000 0040                  OP442 - 1 = MPCR                           06530000 D
13FF  10B0 0000 0000 0040                  OP443 - 1 = MPCR                           06531000 D
                                                          %                           06532000 D
                                                          %                           06533000 D
                                   OP440:                 % RR TYPE JUMP ZERO          06534000 D
                                                          % IF (R(A)) = 0, (R(M)) IN P 06535000 D
1400  22BC 0000 0000 0060          CONTENTSRA - 1 = CPCR  % RETURNS (R(A)) IN B       06536000 D
1401  4809 0052 0030 00F0          B EQL 0                                            06537000 D
```

```
1402  6819 0000 0000 00F0          IF TRUE THEN SKIP                                  06538000 D
1403  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                  06539000 D
1404  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % RETURNS (R(M)) IN B       06540000 D
1405  4809 A000 C000 C0FC          A1 R = A1              % CLEAR OLD PAR              06541000 D
1406  0000 0000 0000 0020          COMP 16 = SAR                                      06542000 D
1407  4809 A001 4000 00F0          A1 L = A1                                          06543000 D
1408  4809 AC5C 4000 00FC          A1 OR B = A1           % CREATE NEW PAR             06544000 D
1409  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                  06545000 D
                                                          %                           06546000 D
                                                          %                           06547000 D
                              OP441:                      % RI TYPE 1 LOCAL JUMP EQUAL 06548000 D
                                                          % IF (CC) = 0R 0, (P) + D INTO P 06549000 D
140A  2250 0000 0000 0060          CHECKCC - 1 = CPCR     % RETURNS CC BITS IN B       06550000 D
140B  4809 0052 0000 00F0          B EQL 0                                            06551000 D
140C  6819 0000 0000 00F0          IF TRUE THEN SKIP                                  06552000 D
140D  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                  06553000 D
140E  5590 0000 0000 0040          RI1 - 1 = MPCR         % (P) + D = P               06554000 D
                                                          %                           06555000 D
                                                          %                           06556000 D
                              OP442:                      % RK TYPE JUMP ZERO          06557000 D
                                                          % IF (R(A)) 0, Y INTO P     06558000 D
140F  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR  % GET (R(A)) INB            06559000 D
1410  4809 0052 0000 00F0          B EQL 0                                            06560000 D
1411  6819 0000 0000 00F0          IF TRUE THEN SKIP                                  06561000 D
1412  1030 0000 0000 0040          BUMP - 1 = MPCR        % ADVANCE THE PAR BY 1 AND CALL OPCODE 06562000 D
1413  4809 E156 1000 00F0          A3 AND LIT = A3        % "M" FIELD IN A3            06563000 D
1414  0050 0000 0000 00EC          15 = LIT                                           06564000 D
1415  4809 E012 0000 C0FC          A3 EQL 0                                           06565000 D
1416  6C19 0000 0000 00FC          IF TRUE THEN 0 = B; SKIP                           06566000 D
1417  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % RETURN (R(M)) IN B        06567000 D
1418  4809 0040 1000 00F0          B = A3                                             06568000 D
1419  633C 0000 0000 0060          IFETCH - 1 = CPCR      % Y INTO B                  06569000 D
141A  4809 EC40 0800 00FC          A3 + B = B             % PAR VALUE IN LS WORD OF E  06570000 D
141B  4809 A000 C000 C0FC          A1 R = A1              % CLEAR OLD PAR              06571000 D
141C  0000 0000 0000 0020          COMP 16 = SAR                                      06572000 D
141D  4809 A001 4000 00F0          A1 L = A1                                          06573000 D
141E  4809 AC5C 4000 00F0          A1 OR B = A1           % CREATE NEW PAR             06574000 D
141F  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                  06575000 D
                                                          %                           06576000 D
                                                          %                           06577000 D
                              OP443:                      % RX TYPE JUMP ZERO          06578000 D
                                                          % IF (R(A)) = 0, (Y) INTO P 06579000 D
1420  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B             06580000 D
1421  4809 0052 0000 00F0          B EQL 0                                            06581000 D
1422  6819 0000 0000 00F0          IF TRUE THEN SKIP                                  06582000 D
1423  1030 0000 0000 0040          BUMP - 1 = MPCR        % ADVANCE THE PAR BY 1 AND CALL OPCODE 06583000 D
1424  4809 E000 0800 00F0          A3 = B                                             06584000 D
1425  5700 0000 0000 0060          RXMFIELD - 1 = CPCR    % ANALYZE THEN "M" FIELD    06585000 D
1426  4809 0041 0010 00FC          B L = BR2              % PUT Y INTO BR2             06586000 D
1427  0000 0000 0000 0030          COMP 8 = SAR                                       06587000 D
1428  60E0 0000 0000 0060          EMULIN - 1 = CPCR      % (Y) INTO B                06588000 D
1429  4809 A000 C000 C0FC          A1 R = A1              % CLEAR OLD PAR              06589000 D
142A  0000 0000 0000 0020          COMP 16 = SAR                                      06590000 D
142B  4809 A001 4000 00F0          A1 L = A1                                          06591000 D
142C  4809 AC5C 4000 00F0          A1 OR B = A1           % CREATE NEW PAR             06592000 D
142D  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                  06593000 D
                                                          %                           06594000 D
                                                          %                           06595000 D
                              OPCODE45:                   % JUMP NOT ZERO              06596000 D
142E  5F90 0000 0000 0060          XFCODE - 1 = CPCR      % RETURN "F" FIELD IN LSD OF A2 06597000 D
```

```
142F  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                  06598000 D
1430  1DC0 00C0 0090 00C0          OP45F - 1 = AMPCR                                   06559000 D
1431  4809 0000 0030 C0F0          STEP                                                06600000 D
1432  4824 00C0 0090 C0F0          EXEC                                                06601000 D
                                                        %                               06602000 D
                                                        %                               06603000 D
1433  1D00 0000 0000 0040    OP45F:  OP450 - 1 = MPCR                                  06604000 D
1434  1DE0 00C0 0090 004C          OP451 - 1 = MPCR                                    06605000 D
1435  1DF0 0000 0070 C040          OP452 - 1 = MPCR                                    06606000 D
1436  1E00 C000 0C30 C040          OP453 - 1 = MPCR                                    06607000 D
                                                        %                               06608000 D
                                                        %                               06609000 D
                            OP450:                      % RR TYPE JUMP NOT ZERO          06610000 D
                                                        % IF (R(A)) NE 0, (R(M)) INTO P 06611000 D
1437  2280 0000 0030 006C          CONTENTSRA - 1 = CPCR   % (R(A)) INTO B             06612000 D
1438  4809 0C52 0000 00F0          B EQL 0                                             06613000 D
1439  680E 00C0 0030 00F0          IF TRUE THEN STEP ELSE SKIP                         06614000 D
143A  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                   06615000 D
143B  2380 0000 0030 0060          CONTENTSRM - 1 = CPCR  % RETURN (R(M)) IN B         06616000 D
143C  4809 A000 0090 00F0          A1 R = A1                                           06617000 D
143D  0000 0000 0090 0020          COMP 16 = SAR                                       06618000 D
143E  4809 A0C1 4000 C0F0          A1 L = A1               % A1 PAR FIELD CLEARED       06619000 D
143F  4809 AC5C 4000 C0FC          A1 OR B = A1            % CREATE NEW PAR             06620000 D
1440  66E0 C000 C000 C040          OPCODE - 1 = MPCR                                   06621000 D
                                                        %                               06622000 D
                                                        %                               06623000 D
                            OP451:                      % RI TYPE JUMP NOT EQUAL         06624000 D
                                                        % IF (CC) NE, (P) + D INTO P     06625000 D
1441  2260 00C0 0C00 0060          CHECKCC - 1 = CPCR      % PUT CC BITS IN B           06626000 D
1442  4809 0C52 0030 0CF0          B EQL 0                                             06627000 D
1443  6806 0000 0090 C0F0          IF TRUE THEN STEP ELSE SKIP                         06628000 D
1444  66E0 00C0 0C30 0040          OPCODE - 1 = MPCR                                   06629000 D
1445  3590 0000 0000 C040          RI1 - 1 = MPCR          % (P) + D = P                06630000 D
                                                        %                               06631000 D
                                                        %                               06632000 D
                            OP452:                      % RK TYPE JUMP NOT ZERO          06633000 D
                                                        % IF (R(A)) NE 0, Y INTO P       06634000 D
1446  2280 00C0 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B              06635000 D
1447  4809 0C52 0030 C0F0          B EQL 0                                             06636000 D
1448  680E 00C0 0C30 C0F0          IF TRUE THEN STEP ELSE SKIP                         06637000 D
1449  1030 C0C0 0000 0040          BUMP - 1 = MPCR         % ADVANCE THE PAR BY 1 AND CALL OPCODE 06638000 D
144A  4809 E155 1090 C0F0          A3 AND LIT = A3         % "M" FIELD IN A3           06639000 D
144B  00FC 00C0 0070 00E0          15 = LIT                                            06640000 D
144C  4809 EC12 0090 00FC          A3 EQL 0                                            06641000 D
144D  6C19 00C0 0B00 00F0          IF TRUE THEN 0 = B; SKIP                            06642000 D
144E  2380 0000 0090 0060          CONTENTSRM - 1 = CPCR  % (R(M)) IN B                06643000 D
144F  4809 CC40 1090 00FC          P = A3                                              06644000 D
1450  533C C0C0 0000 C060          IFETCH - 1 = CPCR       % PUT Y INTO B              06645000 D
1451  4809 EC40 0B00 00F0          A3 + B = B                                          06646000 D
1452  4809 A0C0 C030 00F0          A1 R = A1                                           06647000 D
1453  0000 00C0 0000 002C          COMP 16 = SAR                                       06648000 D
1454  4809 A0C1 4000 C0FC          A1 L = A1                                           06649000 D
1455  4809 AC5C 4000 00F0          A1 OR B = A1            % CREATE NEW PAR             06650000 D
1456  66E0 00C0 0C30 0040          OPCODE - 1 = MPCR                                   06651000 D
                                                        %                               06652000 D
                                                        %                               06653000 D
                            OP453:                      % RX TYPE JUMP NOT ZERO          06654000 D
                                                        % IF (R(A)) NE 0, (Y) INTO P     06655000 D
1457  4809 CC40 8B00 00F0          B R = B                 % SHIFT "A" TO LS BITS        06656000 D
1458  8CFC 00C0 0000 C080          Q = SAR; 15 = LIT                                   06657000 D
```

```
1459  4809 2C56 0600 00F0          LIT AND B = B          % ISOLATE "A"                    06658C00 D
145A  51CC 0CC0 0000 C060          REGSTACK - 1 = CPCR     % R(A) INTO MAR2                06659C00 D
145B  2F30 0CC0 0C0C C060          EINPUT - 1 = CPCR      % (R(A)) INTO B                  06660C00 D
145C  4809 0C52 0C0C 00F0          B EQL 0                                                 06661C00 D
145D  580B 0C00 0030 00FC          IF TRUE THEN STEF ELSE SKIP                             06662C0C D
145E  103C 0000 0030 0040          BUMP - 1 = MPCR        % ADVANCE THE PAR BY 1 AND CALL OPCODE 06663000 D
145F  57D0 00C3 0C30 C060          RXMFIELD - 1 = CPCR    % ANALYZE "M" FIELD              06664C0C D
1460  4809 0C41 0010 00FC          D L = BR2              % ADDRESS OF Y INTO BR2           06665C0C D
1461  0000 C0C0 0000 C03C          COMP B = SAR                                            06666C00 D
1462  60EC C000 0C0C C060          EMULIN - 1 = CPCR      % (Y) IN B                       06667C00 D
1463  48C9 A0C0 C000 00F0          A1 R = A1                                               06668C00 D
1464  000C 00C0 0030 0020          COMP 16 = SAR                                           06669C00 D
1465  4809 A0C1 4030 C0F0          A1 L = A1                                               0667CC00 D
1466  4809 AC5C 4000 00F0          A1 OR B = A1           % CREATE NEW PAR                 06671C00 D
1467  66E0 C0C3 0C00 C040          OPCODE - 1 = MPCR                                       06672C00 D
                                                          %                                06673C00 D
                                                          %                                06674C00 D
                                   OPCODE46:              % JUMP POSITIVE                   06675C00 D
1468  5F9C 00C0 000C 00B0          XFCODE - 1 = CPCR      % GET "F" CODE IN A2              06676C00 D
1469  4809 C640 0040 00FC          A2 + AMPCR = AMPCR                                      06677C00 D
146A  1E10 00C3 0C00 00C0          OP46F - 1 = AMPCR                                       06678C0C D
146B  4809 0000 0030 00FC          STEP                                                    06679C00 D
146C  4824 C003 0C0C C0F0          EXEC                                                    06680C00 D
                                                          %                                06681C00 D
                                                          %                                06682C00 D
146D  1E20 0000 0C00 0040          OP46F:  OP460 - 1 = MPCR                                06683C00 D
146E  1E3C 00C3 0C00 0040          OP461 - 1 = MPCR                                        06684C00 D
146F  1E40 00C0 0030 0040          OP462 - 1 = MPCR                                        06685C00 D
1470  1E50 0000 0000 C040          OP463 - 1 = MPCR                                        06686C00 D
                                                          %                                06687C0C D
                                                          %                                06688C00 D
                                   OP460:                 % RR TYPE JUMP POSITIVE           06689C00 D
                                                          % IF (R(A)) >= 0, (R(M)) INTO P  06690C00 D
1471  228C 00C0 0C0C C060          CONTENTSRA - 1 = CPCR % (R(A)) INTO B                   06691C00 D
1472  4809 CC41 0800 00F0          B L = B                % SHIFT (R(A)) TO MS WORD         06692C00 D
1473  300C C0C3 0000 0020          COMP 16 = SAR                                           06693C00 D
1474  4809 0C40 0C0C C0F0          B                      % PUT B IN THE ADDER              06694C00 D
1475  480E 00C0 0000 00F0          IF MST THEN STEP ELSE SKIP  % TEST (R(A)) >= 0          06695C00 D
1476  66E0 00C0 0030 0040          OPCODE - 1 = MPCR                                       06696C00 D
1477  2380 0000 0000 C060          CONTENTSRM - 1= CPCR   % (R(M)) INTO B                  06697C00 D
1478  4809 A0C0 C000 00FC          A1 R = A1              % CLEAR THE PAR FIELD             06698C00 D
1479  3000 00C3 0030 0020          COMP 16 = SAR                                           06699C00 D
147A  4809 A0C1 4030 C0F0          A1 L = A1                                               0670CC00 D
147B  4809 AC5C 4000 00F0          A1 OR B = A1           % SET UP NEW PAR                 06701C00 D
147C  66E0 C000 C000 0040          OPCODE - 1 = MPCR                                       06702C00 D
                                                          %                                06703C00 D
                                                          %                                06704C00 D
                                   OP461:                 % RI TYPE 1 LOCAL JUMP >=         06705C00 D
147D  226C 0000 0000 C06C          CHECKCC - 1 = CPCR     % RETURN CC IN LS BITS OF E      06706C00 D
147E  4809 2C52 0C0C C0F0          LIT EQL B              % TEST FOR <                     0670CC00 D
147F  0030 0CC3 0030 C0EC          3 = LIT                                                 06708C00 D
1480  68CB 0000 C000 CCFC          IF TRUE THEN STEF ELSE SKIP                             06709000 D
1481  66EC 0000 0030 0040          OPCODE - 1 = MPCR                                       0671CC00 D
1482  559C 0000 0000 0040          RI1 - 1 = MPCR         % (P) + D = P                    06711C00 D
                                                          %                                06712CC0 D
                                                          %                                06713C00 D
                                   OP462:                 % RK TYPE JUMP POSITIVE           06714C00 D
                                                          % IF (R(A)) >= 0, Y INTO P       06715C0C D
1483  2280 CCC3 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                  06716C00 D
1484  4809 CC41 0B00 00F0          B L = B                % SHIFT (R(A)) INTO MS WORE OF B  06717C0C D
```

```
1485  0000 0000 0000 0020          COMP 16 = SAR                                  06718000 D
1486  4809 0C40 0000 00F0          B                    % PUT B IN THE ADDER      06719000 D
1487  480E 0C00 0000 00F0          IF MST THEN STEP ELSE SKIP                    06720000 D
1488  1D30 0000 0000 0040          BUMP - 1 = MPCR      % ADVANCE THE PAR BY 1 AND CALL OPCODE 06721000 D
1489  4809 E156 1000 00F0          A3 AND LIT = A3      % "M" FIELD IN A3         06722000 D
148A  00FC 0000 0000 00E0          15 = LIT                                      06723000 D
148B  4809 E012 0000 00F0          A3 EOL 0                                      06724000 D
148C  6C19 0000 0000 00FC          IF TRUE THEN 0 = B; SKIP                      06725000 D
148D  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO 9       06726000 D
148E  4809 0C40 1000 00F0          0 = A3                                        06727000 D
148F  633C 0000 0000 C060          IFETCH - 1 = CPCR    % Y INTO B               06728000 D
1490  4809 EC40 C000 00F0          A3 + B = B           % PUT PAR IN LS WORD OF B 06729000 D
1491  4809 A000 0000 00F0          A1 R = A1                                     06730000 D
1492  0000 0000 0000 C020          COMP 16 = SAR                                 06731000 D
1493  4809 A0C1 4000 00F0          A1 L = A1            % CLEAR PAR FROM PSW      06732000 D
1494  4809 AC5C 4000 00F0          A1 OR B = A1                                  06733000 D
1495  66EC 0000 0000 C040          OPCODE - 1 = MPCR                             06734000 D
                                                        %                         06735000 D
                                                        %                         06736000 D
                                OP463:                  % RX JUMP POSITIVE        06737000 D
                                                        % IF (R(A)) >= 0, (Y) INTO P 06738000 D
1496  4809 0C40 8800 00F0          B R = B              % "A" INTO LS BITS        06739000 D
1497  80FC 0000 0000 C080          4 = SAR; 15 = LIT                             06740000 D
1498  4809 2C56 0800 00F0          LIT AND B = B        % ISOLATE "A" INTO B      06741000 D
1499  51C0 0000 0000 C060          REGSTACK - 1 = CPCR  % R(A) INTO MAR2         06742000 D
149A  2F3C 0000 0000 C060          EINPUT - 1 = CPCR    % (R(A)) INTO E          06743000 D
149B  4809 0C41 0800 00F0          B L = B              % SHIFT (R(A)) INTO MS WORD OF B 06744000 D
149C  0000 0000 0000 C020          COMP 16 = SAR                                 06745000 D
149D  4809 0C40 0000 00F0          B                    % PUT B IN THE ADDER      06746000 D
149E  480B 0000 0000 00F0          IF MST THEN STEP ELSE SKIP                    06747000 D
149F  1D30 0000 0000 C040          BUMP - 1 = MPCR      % ADVANCE THE PAR BY 1 AND CALL OPCODE 06748000 D
14A0  4809 E0C0 0800 00F0          A3 = B                                        06749000 D
14A1  5700 0000 0000 0060          RXMFIELD - 1 = CPCR                           06750000 D
14A2  4809 0C41 0010 00F0          B L = BR2            % PUT Y INTO BR2          06751000 D
14A3  0000 0000 0000 C030          COMP 8 = SAR                                  06752000 D
14A4  60EC 0000 0000 0060          EMULIN - 1 = CPCR    % PUT (Y) INTO B         06753000 D
14A5  4809 A000 0000 00FC          A1 R = A1            % CLEAR OLD PAR           06754000 D
14A6  0000 0000 0000 C020          COMP 16 = SAR                                 06755000 D
14A7  4809 A001 4000 00FC          A1 L = A1                                     06756000 D
14A8  4809 AC5C 4000 00FC          A1 OR B = A1         % CREATE NEW PAR          06757000 D
14A9  66EC 0000 0000 0040          OPCODE - 1 = MPCR                             06758000 D
                                                        %                         06759000 D
                                                        %                         06760000 D
                                OPCODE47:               % JUMP NEGATIVE INSTRUCTION 06761000 D
14AA  5F9C 0000 0000 C060          XFCODE - 1 = CPCR    % RETURNS "F" FIELD IN LSB OF A2 06762000 D
14AB  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                            06763000 D
14AC  1E6C 0000 0000 00C0          OP47F - 1 = AMPCR                             06764000 D
14AD  4809 0000 0000 00F0          STEP                                          06765000 D
14AE  4824 0000 0000 00FC          EXEC                 % JUMP TO DESIRED FORMAT TYPE 06766000 D
                                                        %                         06767000 D
                                                        %                         06768000 D
14AF  1E70 0000 0000 0040  OP47F:  OP470 - 1 = MPCR                              06769000 D
14B0  1E8C 0000 0000 0040          OP471 - 1 = MPCR                              06770000 D
14B1  1E9C 0000 0000 0040          OP472 - 1 = MPCR                              06771000 D
14B2  1EA0 0000 0000 0040          OP473 - 1 = MPCR                              06772000 D
                                                        %                         06773000 D
                                                        %                         06774000 D
                                OP470:                  % FR FORMAT JUMP REGISTER NEGATIVE 06775000 D
                                                        % IF (R(A)) < (, (R(M)) INTO P 06776000 D
1483  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B        06777000 D
```

```
14B4  4809 0C41 0C30 C0F0          B L = B                  % SHIFT (R(A)) INTO UHW OF B          06778C0C D
14B5  0000 0000 0000 C02C          COMP 16 = SAR                                                 06779C0C D
14B6  4809 0C40 0000 00F0          B                        % PUT B IN THE ADDER, TEST (R(A)) < 0 06780C00 D
14B7  4819 0000 0000 00FC          IF MST THEN SKIP                                              06781C00 D
14B8  66EC 0000 0000 C040          OPCODE - 1 = MPCR                                             06782C00 D
14B9  2380 0000 0000 C06C          CONTENTSRM - 1 = CPCR    % (R(M)) INTO B                      06783C00 D
14BA  4809 ACC0 C000 00F0          A1 R = A1                % CLEAR LOWER 16 BITS OF A1           06784C00 D
14BB  0000 00C0 0000 0020          COMP 16 = SAR                                                 06785C00 D
14BC  4809 ACC1 4000 0010          A1 L = A1                                                     06786C00 D
14BD  4809 AC5C 4000 C0F0          A1 OR B = A1             % MODIFY PAR TO JUMP                  06787C00 D
14BE  56EC 0000 0000 C040          OPCODE - 1 = MPCR                                             06788C0C D
                                                            %                                     06789C00 D
                                                            %                                     06790C00 D
                                   OP471:                   % RI TYPE I LOCAL JUMP LESS INSTRUCTION 06791C00 D
                                                            % IF (CC) = 11, (P) + D = P            06792C00 D
14BF  2809 0000 0000 C0F0          IF LC1                                                        06793C0C D
14C0  2260 0000 0000 0060          CHECKCC - 1 = CPCR       % PUT THE CC BITS INTO B              06794C00 D
14C1  4809 2C52 0000 C0F0          LIT EQL B                % CHECK CC FOR 11                     06795C00 D
14C2  003C 0000 0000 00E0          3 = LIT                                                       06796C00 D
14C3  6CCB 0000 0000 C0F0          IF FALSE THEN STEP ELSE SKIP                                  06797C00 D
14C4  66EC 0000 0000 0040          OPCODE - 1 = MPCR                                             06798C00 D
14C5  5590 0000 0000 00C0          RI1 - 1 = MPCR           % (P) + D = P                        06799C00 D
                                                            %                                     06800000 D
                                                            %                                     06801C0C D
                                   OP472:                   % RK TYPE JUMP NEGATIVE               06802000 D
                                                            % IF (R(A)) < 0, (Y) = P              06803C0C D
14C6  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR    % (R(A)) INTO B                      06804C00 D
14C7  4809 0C41 CB30 C0F0          B L = B                  % SHIFT (R(A)) INTO UHW OF B          06805C00 D
14C8  00C0 00C0 0000 0020          COMP 16 = SAR                                                 06806C00 D
14C9  4809 0C40 0000 C0FC          B                        % CHECK IF (R(A)) < 0                 06807C00 D
14CA  4819 00C0 0000 C0F0          IF MST THEN SKIP                                              06808C00 D
14CB  1030 00C0 0000 C040          BUMP - 1 = MPCR          % ADVANCE THE PAR BY 1 AND CALL OPCODE 06809C00 D
14CC  2380 60C0 0000 C060          CONTENTSRM - 1 = CPCR    % (R(M)) INTO E                      06810C00 D
14CD  4809 0C40 1000 00F0          B = A3                                                        06811C00 D
14CE  633C 0000 0000 0060          IFETCH - 1 = CPCR        % PUT Y INTO B                       06812C00 D
14CF  4809 EC40 0800 00F0          A3 + B = B               % PUT PAR VALUE IN LS WORD OF B       06813C00 D
14D0  4809 A0C0 C000 00FC          A1 R = A1                % CLEAR OLD PAR                       06814C00 D
14D1  0000 00C0 0000 002C          COMP 16 = SAR                                                 06815C00 D
14D2  4809 A001 4000 00FC          A1 L = A1                                                     06816C00 D
14D3  4809 AC5C 4030 C0F0          A1 OR B = A1             % CONSTRUCT NEW PAR                   06817C00 D
14D4  56EC 00C0 0000 0040          OPCODE - 1 = MPCR                                             06818C00 D
                                                            %                                     06819C00 D
                                                            %                                     06820C00 D
                                   OP473:                   % RX TYPE JUMP INSTRUCTION            06821C00 D
                                                            % IF (R(A)) < 0, (Y) = P              06822C00 D
14D5  4809 0C40 8B30 C0FC          B R = B                  % SHIFT "A" INTO LS BITS OF B         06823C00 D
14D6  90F0 00C0 0030 0080          4 = SAR; 15 = LIT                                             06824C00 D
14D7  4809 2C56 0B30 C0F0          LIT AND B = B            % ISOLATE "A" INTO B                 06825C00 D
14D8  51C0 0000 C000 C060          REGSTACK - 1 = CPCR      % R(A) INTO MAR2                     06826C00 D
14D9  2F30 0000 0000 0060          EINPUT - 1 = CPCR        % (R(A)) INTO E                      06827C00 D
14DA  4809 0C41 0B00 00F0          B L = B                  % (R(A)) SHIFTED INTO UHW OF B        06828C00 D
14DB  0000 0000 0C00 C02C          COMP 16 = SAR                                                 06829C00 D
14DC  4809 0C40 0030 C0FC          B                        % CHECK IF (R(A)) < 0                 06830C00 D
14DD  4819 00C0 0030 00F0          IF MST THEN SKIP                                              06831C00 D
14DE  1030 0000 C000 C040          BUMP - 1 = MPCR          % ADVANCE THE PAR BY 1 AND CALL OPCODE 06832C00 D
14DF  4809 E0C0 0B30 C0FC          A3 = B                   % RESTORE INSTRUCTION IN B           06833C00 D
14E0  57CC 00C0 0030 C060          RXMFIELD - 1 = CPCR      % ANALYZE "M" FIELD                  06834C00 D
14E1  4809 0C41 CC10 C0F0          E L = BR2                % PUT Y INTO BR2                      06835C00 D
14E2  0000 00C0 0000 0030          COMP 8 = SAR                                                  06836C00 D
14E3  56EC 00C0 0000 C060          EMULIN - 1 = CPCR        % PUT (Y) INTO B                     06837C00 D
```

```
14E4  4809 A003 C030 00F0          A1 R = A1                  % CLEAR THE OLD PAR          06838C0G D
14E5  0000 0003 0030 0020          COMP 16 = SAR                                           06839C00 D
14E6  4809 A001 4000 00F0          A1 L = A1                                               06840C00 D
14E7  4809 AC5C 4000 00F0          A1 OR B = A1               % PUT (Y) INTO THE PAR        06841C00 D
14E8  66EC 0000 0000 0C40          OPCODE - 1 = MPCR                                       06842C00 D
                                                              %                            06843C00 D
                                                              %                            06844C00 C
                                OPCODE50:                     % NOT IMPLEMENTED             06845000 D
14E9  4E50 0000 0030 0040          NOTIMP - 1 = MPCR                                       06846C00 D
                                                              %                            C6847C0G D
                                OPCODE51:                     % NOT IMPLEMENTED             06848C00 D
14EA  4E5C 0000 0030 0040          NOTIMP - 1 = MPCR                                       06849C00 D
                                                              %                            06850C00 D
                                OPCODE52:                     % NOT IMPLEMENTED             06851000 D
14EB  4E50 0003 0030 004C          NOTIMP - 1 = MPCR                                       C6852C0G D
                                                              %                            C685300C D
                                OPCODE53:                     % NOTIMPLEMENTED              C6854C00 D
14EC  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                                       C6855C00 D
                                                              %                            C6856C0C D
                                                              %                            06857C00 D
                                OPCODE54:                     % LOAD ADDRESS REGISTER(S)    06858C00 D
14ED  5F90 0003 0030 C060          XFCODE - 1 = CPCR                                       06859C00 D
14EE  4809 C640 0C40 00F0          A2 + AMPCR = AMPCR                                      C6860C00 D
14EF  1EBC 0003 C030 00C0          OP54F - 1 = AMPCR                                       06861C00 D
14F0  4809 0000 0030 00FC          STEP                                                    06862C00 D
14F1  4824 0000 0000 00F0          EXEC                                                    C6863C00 D
                                                              %                            06364C00 D
14F2  1ECC 0003 0030 0040       OP54F:  OP540 - 1 = MPCR                                   06865C00 D
14F3  1ED0 0003 0030 0040          OP541 - 1 = MPCR                                        06866C00 D
14F4  300C 0003 0030 0040          FAULT - 1 = MPCR                                        06867C00 D
14F5  1EE0 0000 0000 0040          OP543 - 1 = MPCR                                        06868C00 D
                                                              %                            C6869C0C D
                                OP540:                        % LARR; LOAD ADDRESS REGS; TYPE RR 06870C00 D
14F6  4809 E155 0B00 00F0          A3 AND LIT = B                                          06871C00 D
14F7  00FC 0003 0000 00E0          15 = LIT                                                06872C00 D
14F8  51C0 0003 0000 006C          REGSTACK - 1 = CPCR                                     06873C00 D
14F9  2F30 0003 0000 0060          EINPUT - 1 = CPCR          % B = (R(M))                 06874C00 D
14FA  4809 0C40 0030 C0F0          B = MIR                                                 06875C0C D
                                                              %                            06876C0C D
14FB  4809 E003 8B00 00F0       LAR:   A3 R = B                                            06877C00 D
14FC  30F0 0000 0C00 0080          4 = SAR; 15 = LIT                                       06878C0C D
14FD  4809 2C56 CB00 CCF0          LIT AND B = B              % B = R(A)                   06879C00 D
14FE  51C0 0003 0000 0060          REGSTACK - 1 = CPCR                                     C688CC0C D
14FF  2F30 0003 000C 0C6C          EINPUT - 1 = CPCR          % B = (R(A))                 06881C00 D
1500  48C9 2C56 2000 00F0          LIT AND B = A2             % A2 = (R(A))7-0              06882C00 D
1501  0FFC 0003 0030 00E0          255 = LIT                  % 8 BIT MASK                  06883C00 D
1502  4809 C640 001C C0F0          A2 + AMPCR = MAR2          % MAR2 = AR                   06884C00 D
1503  3000 0000 0000 00C0          PAGEREG = AMPCR            % ADD EASE ADDR + A2          06885C00 D
1504  4809 0000 0030 C0F0          STEP                       % SEPARATE AMPCR ASSIGNMENTS  06886C0C D
1505  2F50 0003 0030 C06C          EOUTPUT - 1 = CPCR                                      06887C0C D
1506  66EC 0003 0030 0040          OPCODE - 1 = MPCR                                       06888C00 D
                                                              %                            06889C00 D
                                OP541:                        % LARI; LOAD ADDRESS REGISTER (INDIRECT) 06890C0C D
                                                              % (Y*) -> AR R                06891C00 D
1507  4809 E156 0B00 C0F0          A3 AND LIT = B                                          06892C00 D
1508  00FC 0003 0000 C0EC          15 = LIT                                                06893C00 D
1509  51C0 0003 0030 0060          REGSTACK - 1 = CPCR                                     06894C00 D
150A  2F30 0003 0030 0060          EINPUT - 1 = CPCR          % B = (R(M)) = Y*            06895C00 D
150B  4809 0C41 0010 C0F0          B L = BR2                  % Y* INTO BR2                 06896C00 D
150C  0000 0003 0030 0030          COMP 8 = SAR                                            06897C00 D
```

```
1500  60EC 0000 0000 0060          EMULIN - 1 = CPCR      % B = (Y*)                    06898000 D
150E  4809 0C40 0090 00FC          B = MIR                                              06899000 D
150F  4FA0 0000 0000 0050          LAR - 1 = MPCR                                       06900000 D
                                                          %                             06901000 D
                                OP543:                    % LARM; LOAD ADDRESS REG. MULTIPLE 06902000 D
                                                          %(Y,...Y+U) -> AR R...AR R+U  06903000 D
                                                          % MULTIPLE LOADS OBSERVE PAGE 06904000 D
                                                          % ADDRESSES                   06905000 D
1510  57C0 0000 0000 0060          RXMFIELD - 1 = CPCR    % B = Y                       06906000 D
1511  4809 0C40 0090 00F0          B = MIR                                              06907000 D
1512  4809 E000 8800 00F0          A3 R = B                                             06908000 D
1513  80FC 0000 0000 0080          4 = SAR; 15 = LIT                                    06909000 D
1514  4809 2056 0800 00F0          LIT AND B = B          % E = :A: FIELD               06910000 D
1515  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                                  06911000 D
1516  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % B = (R(A))                  06912000 D
1517  0FF0 0000 0000 0090          255 = LIT ; B = SAR    % A2 = PAGE REG NUM.           06913000 D
1518  4809 2056 2000 00F0          LIT AND B = A2                                       06914000 D
1519  4809 C640 1000 00F0          A2 + AMPCR = A3        % A3 = AR R                    06915000 D
151A  3000 0000 0000 0000          PAGEREG = AMPCR                                      06916000 D
1518  49C9 0C40 8800 00F0          B R = B; SET LC1       % B HAS COUNT; LC1 SET FOR R +1 MOVE 06917000 D
151C  4809 0C52 0000 00F0          C EQL B                % COUNT OF ZERO CAUSES ALL REG TO BE 06918000 D
1510  9FFC 0000 0000 00A0          255 = LIT; COMP 16 = SAR                             06919000 D
151E  6C19 2001 0800 00F0          IF TRUE THEN LIT + 1 L = B; EKIP                     06920000 D
151F  4809 0C47 0800 00F0          B + 1 L = B                                          06921000 D
1520  4809 EC5C 1000 00F0          A3 OR B = A3,BM1       % COUNT IN UHW OF A3          06922000 D
1521  4809 0C41 0020 00F0          B L = BR1              % ORIGIN ADDR IN BR1           06923000 D
1522  0000 0000 0000 0030          COMP B = SAR                                         06924000 D
1523  4A20 0000 0000 0060          MOVE - 1 = CPCR                                      06925000 D
1524  66EC 0000 0000 0040          OPCODE - 1 = MPCR                                    06926000 D
                                                          %                             06927000 D
                                                          %                             06928000 D
                                OPCODE55:                 % STORE ADDRESS REGISTER(S)   06929000 D
1525  5F90 0000 0000 0060          XFCODE - 1 = CPCR                                    06930000 D
1526  4809 C640 0040 00FC          A2 + AMPCR = AMPCR                                   06931000 D
1527  1EFC 0000 0000 0000          OP55F - 1 = AMPCR                                    06932000 D
1528  4809 0000 0000 00F0          STEP                                                 06933000 D
1529  4824 0000 0000 00F0          EXEC                                                 06934000 D
                                                          %                             06935000 D
152A  1F00 0000 0000 0040       OP55F:  OP550 - 1 = MPCR                                06936000 D
152B  1F10 0000 0000 0040               OP551 - 1 = MPCR                                06937000 D
152C  3000 0000 0000 0040               FAULT - 1 = MPCR                                06938000 D
152D  1F20 0000 0000 0040               OP553 - 1 = MPCR                                06939000 D
                                                          %                             06940000 D
                                OP550:                    % SARR; STORE ADDRESS REGISTER (REGISTER)06941000 D
                                                          % TYPE RR; (AR) A -> RM       06942000 D
152E  4809 0C40 8800 00F0          B R = B                                              06943000 D
152F  80FC 0000 0000 0080          4 = SAR; 15 = LIT                                    06944000 D
1530  4809 2056 0800 00F0          LIT AND B = B                                        06945000 D
1531  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % MAR2 = RA                   06946000 D
1532  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % B = (R(A))                  06947000 D
1533  4809 2056 2000 00F0          LIT AND B = A2         % A2 = REG NUM.                06948000 D
1534  0FFC 0000 0000 00E0          255 = LIT                                            06949000 D
1535  4809 C640 0010 00F0          A2 + AMPCR = MAR2                                    06950000 D
1536  3000 0000 0000 00C0          PAGEREG = AMPCR        % PAGEREG BASE + REG #        06951000 D
1537  4809 0000 0000 00FC          STEP                   % SEPARATE AMPCR ASSIGNMENTS  06952000 D
1538  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % B = (AR)                    06953000 D
1539  4809 0C40 0030 00F0          B = MIR                                              06954000 D
153A  4809 E156 0800 00F0          A3 AND LIT = B         % B = :M: FIELD               06955000 D
153B  0CFC 0000 0000 00E0          15 = LIT                 \                           06956000 D
153C  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % MAR2 = RM                   06957000 D
```

```
153D  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR     % (AR) -> RM                    06958000 D
153E  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                      06959000 D
                                                          %                               06960000 D
                                   OP551:                 % SARI; STORE ADDRESS REGISTER   06961000 D
                                                          % TYPE RI; (AR) A -> Y*         06962000 D
153F  4809 0040 8B00 00F0          6 R = B                                                06963000 D
1540  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                      06964000 D
1541  4809 2C56 0B00 00F0          LIT AND B = B          % B = :A: FIELD                 06965000 D
1542  51C0 0000 0000 0060          REGSTACK - 1 = CPCR    % MAR2 = RA                     06966000 D
1543  2F30 0000 0030 0060          EINPUT - 1 = CPCR      % B = (R(A))                    06967000 D
1544  4809 2C56 2000 00F0          LIT AND B = A2         % A2 = REG 0                    06968000 D
1545  0FF0 0000 0000 00E0          255 = LIT                                              06969000 D
1546  4809 C640 001C 00F0          A2 + AMPCR = MAR2                                      06970000 D
1547  3000 0000 0000 0000          PAGEREG = AMPCR        % PAGEREG BASE + REG 0          06971000 D
1548  4809 0000 0000 00F0          STEP                   % SEPARATE AMPCR ASSIGNMENTS    06972000 D
1549  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % B = (AR)                      06973000 D
154A  4809 0C40 0030 00F0          B = MIR                                                06974000 D
154B  4809 E156 0B00 00F0          A3 AND LIT = B         % B = :M: FIELD                 06975000 D
154C  00F0 0000 0000 00E0          15 = LIT                                               06976000 D
154D  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                                    06977000 D
154E  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % C = (R(M)) = Y*               06978000 D
154F  4809 0C41 0010 00F0          B L = BR2              % Y* INTO BR2                    06979000 D
1550  0000 0000 0000 0030          COMP 8 = SAR                                           06980000 D
1551  51E0 0000 0000 0060          EMULOUT - 1 = CPCR     % (AR) -> Y*                    06981000 D
1552  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                      06982000 D
                                                          %                               06983000 D
                                   OP553:                 % STORE ADDRESS REGISTER MULTIPLE 06984000 D
                                                          % SARM; TYPE RX (AR R,...AR R+U) -> 06985000 D
                                                          % Y,...Y+U                      06986000 D
                                                          % MULTIPLE STORES DO OBSERVE PAGE 06987000 D
                                                          % ADDRESSES                     06988000 D
1553  5700 0000 0000 0060          RXMFIELD - 1 = CPCR                                    06989000 D
1554  4809 0C40 0030 00F0          B = MIR                % MIR = Y                        06990000 D
1555  4809 E000 8B00 00F0          A3 R = B                                               06991000 D
1556  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                      06992000 D
1557  4809 2C56 0B00 00F0          LIT AND B = B          % B = :A: FIELD                 06993000 D
1558  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                                    06994000 D
1559  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % B = (R(A))                    06995000 D
155A  4809 2C56 2000 00F0          LIT AND B = A2         % A2 HAS PAGE REG 0             06996000 D
155B  0FF0 0000 0000 0090          255 = LIT; 8 = SAR                                     06997000 D
155C  4809 C640 1000 00F0          A2 + AMPCR = A3        % P = AR R                      06998000 D
155D  3000 0000 0000 0000          PAGEREG = AMPCR                                        06999000 D
155E  4809 0C40 8B00 00F0          E R = B                % SHIFT COUNT INTO LS BITS OF P  07000000 D
155F  4809 0C52 0000 00F0          0 EQL B                % A COUNT OF ZERO MEANS ALL REG STORED 07001000 D
1560  0FF0 0000 0000 00A0          255 = LIT; COMP 16 = SAR                               07002000 D
1561  6C19 2001 9B00 00F0          IF TRUE THEN LIT + 1 L = B; SKIP                       07003000 D
1562  4809 0C47 0B00 00F0          B + 1 L= B             % COUNT IN B                    07004000 D
1563  4809 E001 0020 00F0          A3 L = BR1             % ORIGIN ADDR IN BR1            07005000 D
1564  0000 0000 0000 0030          COMP 8 = SAR                                           07006000 D
1565  4809 0C40 1000 00F0          B = A3,8MI             % COUNT IN UHW OF A3            07007000 D
1566  4809 EC5C 1000 00F0          A3 OR B = A3           % DESTINATION ADDR IN LHW OF A3 07008000 D
1567  9A20 0000 0000 0060          MOVE - 1 = CPCR                                        07009000 D
1568  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                      07010000 D
                                                          %                               07011000 D
                                                          %                               07012000 D
                                   OPCODE56:              % NOT IMPLEMENTED                07013000 D
1569  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                                      07014000 D
                                                          %                               07015000 D
                                   OPCODE57:              % NOT IMPLEMENTED                07016000 D
156A  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                                      07017000 D
```

```
                                                                                                  %                                                  07018000 D
156B  5F9C 0000 0000 0060        OPCODE60:  XFCODE - 1 = CPCR                                                                                        07019000 D
156C  4809 C640 0030 00F0                   A2 + AMPCR = AMPCR                                                                                       07020000 D
156D  1F30 0000 0030 0000                   OP60F - 1 = AMPCR                                                                                        07021000 D
156E  4809 0000 0030 00F0                   STEP                                                                                                     07022000 D
156F  4824 0000 0000 00F0                   EXEC                                                                                                     07023000 D
                                                                                                  %                                                  07024000 D
1570  1F40 0000 0000 0040        OP60F:     OP600 - 1 = MPCR                                                                                         07025000 D
1571  1F50 0000 0000 0040                   OP601 - 1 = MPCR                                                                                         07026000 D
1572  1F60 0000 0000 0040                   OP602 - 1 = MPCR                                                                                         07027000 D
1573  1F70 0000 0000 0040                   OP603 - 1 = MPCR                                                                                         07028000 D
                                                                                                  %                                                  07029000 D
                                                                                                  %                                                  07030000 D
                                           OP600:                                                 % RL TYPE LOGIC RIGHT SINGLE SHIFT                 07031000 D
                                                                                                  % SHIFT (R(A)) RIGHT M (0-3), SET CC,              07032000 D
1574  22B0 0000 0000 0060                   CONTENTSRA - 1 = CPCR                                  % AND ZEROFILL                                     07033000 D
1575  4809 E156 1000 00F0                   A3 AND LIT = A3                                        % "M" INTO A3                                      07034000 D
1576  00FC 0000 0030 00E0                   15 = LIT                                                                                                 07035000 D
1577  4809 E000 0000 80FC                   A3 = SAR                                                                                                 07036000 D
1578  4809 0040 8030 00F0                   B R = MIR, B                                                                                             07037000 D
1579  2000 0000 0000 0060                   SETCCA - 1 = CPCR                                      % SET THE CONDITION BITS                           07038000 D
157A  2F50 0000 0000 0060                   EOUTPUT - 1 = CPCR                                                                                       07039000 D
157B  66E0 0000 0000 0040                   OPCODE - 1 = MPCR                                                                                        07040000 D
                                                                                                  %                                                  07041000 D
                                                                                                  %                                                  07042000 D
                                           OP601:                                                   % RL TYPE ALG RIGHT SINGLE SHIFT                  07043000 D
                                                                                                  % (R(A)) RIGHT M (0-3), SIGN FILL, SET CC07044000 D
157C  2809 0000 0000 00F0                   IF LC1                                                                                                   07045000 D
157D  22B0 0000 0030 0060                   CONTENTSRA - 1 = CPCR,                                 % (R(A)) INTO B                                   07046000 D
157E  4809 0000 2000 00F0                   0 = A2                                                                                                   07047000 D
157F  4809 0041 0E30 00F0                   B L = B                                                                                                  07048000 D
1580  0006 0000 0000 0020                   COMP 16 = SAR                                                                                            07049000 D
1581  4809 0040 0000 00F0                   B                                                                                                        07050000 D
1582  4009 0018 2000 00F0                   IF MST THEN D111 = A2                                                                                    07051000 D
1583  4809 E156 1000 00F0                   A3 AND LIT = A3                                                                                          07052000 D
1584  00F0 0000 0000 00A0                   15 = LIT; COMP 16 = SAR                                                                                  07053000 D
1585  4809 0001 2000 00F0                   A2 L = A2                                              % A2 = 0000/0000 OR 1111/0000                     07054000 D
1586  4809 0040 8800 00F0                   B R = B                                                                                                  07055000 D
1587  4809 0050 0800 00F0                   A2 OR B = B                                            % B = 1111/(R(A)) OR 0000/(R(A))                  07056000 D
1588  4809 E000 0000 80F0                   A3 = SAR                                                                                                 07057000 D
1589  4809 0040 8800 00F0                   B R = B                                                % PERFORM ALG RIGHT SINGLE SHIFT                   07058000 D
158A  4809 0041 0800 00F0                   B L = B                                                % CLEAR UHW OF B                                   07059000 D
158B  0000 0000 0000 0020                   COMP 16 = SAR                                                                                            07060000 D
158C  4809 0040 8030 00F0                   B R = B, MIR                                                                                             07061000 D
158D  2F50 0000 0000 0060                   EOUTPUT - 1 = CPCR                                                                                       07062000 D
158E  2000 0000 0000 0060                   SETCCA - 1 = CPCR                                      % SET THE CONDITION BITS                           07063000 D
158F  66E0 0000 0000 0040                   OPCODE - 1 = MPCR                                                                                        07064000 D
                                                                                                  %                                                  07065000 D
                                                                                                  %                                                  07066000 D
                                           OP602:                                                 % RL TYPE LOGICAL RIGHT DOUBLE SHIFT               07067000 D
                                                                                                  % (R(A),R(A+1)) M (0-3) PLACES, ZERO FILL07068000 D
                                                                                                  % AND SET CC                                      07069000 D
1590  4809 2056 2000 00F0                   LIT AND B = A2                                         % ISOLATE "M"                                      07070000 D
1591  80F0 0000 0030 0080                   15 = LIT; 4 = SAR                                                                                        07071000 D
1592  4809 0040 8800 00F0                   B R = B                                                                                                  07072000 D
1593  4809 2056 0800 00F0                   LIT AND B = B                                          % OBTAIN "A" FIELD                                07073000 D
1594  51C0 0000 0000 0060                   REGSTACK - 1 = CPCR                                                                                      07074000 D
1595  4809 0F41 0020 00F0                   BMAR L = BR1                                                                                             07075000 D
1596  0000 0000 0000 0030                   COMP B = SAR                                                                                             07076000 D
1597  2F30 0000 0000 0060                   EINPUT - 1 = CPCR                                      % (R(A)) INTO B                                   07077000 D
```

```
1598  4809 0C41 1000 00F0          B L = A3                                  07078C00 D
1599  0010 0000 0000 00A0          16 = SAR ; 1 = LIT                        07079000 D
159A  4809 2F5C 001C 00FC          LIT OR BMAR = MAR2      % R(A+1)          07080C00 D
159B  2F30 0003 0000 C060          EINPUT - 1 = CPCR       % (R(A+1)) INTO B  07081C00 D
159C  4809 EC5C 1000 00F0          A3 OR B = A3            % A3 = (R(A))/(R(A+1))  07082C00 D
159D  4809 CC00 0000 80FC          A2 = SAR                                  07083C00 D
159E  49C9 ED00 9B00 00FC          A3 R = A3,B; SET LC1    % PERFORM DOUBLE SHIFT  07084C00 D
159F  2000 0000 0000 0060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS  07085C00 D
15A0  4809 E001 0B20 00F0          A3 L = B                                  07086C00 D
15A1  0000 0000 0000 C020          COMP 16 = SAR                             07087C00 D
15A2  4809 0C40 903C 00F0          B R = MIR               % NEW (R(A+1))    07088C00 D
15A3  2F50 0003 0000 0060          EOUTPUT - 1 = CPCR      % WRITE NEW R(A+1)  07089C00 D
15A4  4809 0000 0000 20F0          ASR                     % REFERENCE BR1    07090C00 D
15A5  4809 0F40 801C 00FC          BMAR R = MAR2           % R(A)             07091C00 D
15A6  0000 0000 0000 C010          B = SAR                                   07092C00 D
15A7  4809 E003 8030 00F0          A3 R = MIR              % (R(A))           07093C00 D
15A8  0000 0000 0000 002C          16 = SAR                                  07094C00 D
15A9  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR                        07095C00 D
15AA  56E0 0003 0000 0040          OPCODE - 1 = MPCR                         07096C00 D
                                                           %                 07097C00 D
                                                           %                 07098C00 D
                                0F603:                     % RL TYPE ALG RIGHT DOUBLE SHIFT  07099C00 D
                                                           % RIGHT SHIFT (R(A),R(A+1)), M (0-3)  07100C00 D
                                                           % SIGN FILL AND SET CC  07101C00 D
15AB  4809 2C56 2030 C0FC          LIT AND B = A2          % ISOLATE "M" FIELD  07102C00 D
15AC  00F0 0300 0020 C0E0          15 = LIT                                  07103C00 D
15AD  28C9 0C40 8B00 C0FC          B R = B; IF LC1                           07104C00 D
15AE  80FC 0003 0000 0080          4 = SAR; 15 = LIT                         07105C00 D
15AF  4809 2C56 0B00 00F0          LIT AND B = B                             07106C00 D
15B0  51C0 0003 0000 0060          REGSTACK - 1 = CPCR     % R(A) INTO MAR2  07107C00 D
15B1  4809 0F41 0020 00F0          BMAR L = BR1            % TEMP STORAGE    07108C00 D
15B2  0000 0003 0030 0030          COMP 8 = SAR                              07109C00 D
15B3  2F3C 0003 0030 0060          EINPUT - 1 = CPCR       % (R(A)) INTO B   07110C00 D
15B4  4809 0C41 0B00 C0F0          B L = B                                   07111C00 D
15B5  0000 0000 0000 0C2C          COMP 16 = SAR                             07112C00 D
15B6  4809 0C40 0000 C0F0          B                       % PUT B IN THE ADDER  07113C00 D
15B7  4BC9 0C00 0000 C0F0          IF MST THEN SET LC1     % NEG SIGN FLAG   07114C00 D
15B8  0C10 0000 0000 C0A0          COMP 16 = SAR ; 1 = LIT                   07115C00 D
15B9  4809 EC5C 1000 00FC          A3 OR B = A3            % A3 = (R(A))/INSTRUCTION  07116C00 D
15BA  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2      % R(A+1)           07117C00 D
15BB  2F3C 0003 0000 C0A0          EINPUT - 1 = CPCR       % (R(A+1)) INTO B  07118C00 D
15BC  4809 E003 9000 00F0          A3 R = A3               % CLEAR LHW OF A3  07119C00 D
15BD  0000 0000 0000 002C          16 = SAR                                  07120C00 D
15BE  4809 E001 1000 00F0          A3 L = A3                                 07121C00 D
15BF  4809 EC5C 1030 C0F0          A3 OR B = A3            % A3 = (R(A))/(R(A+1))  07122C00 D
15C0  4809 CC00 0000 80F0          A2 = SAR                                  07123C00 D
15C1  4809 E003 9000 00FC          A3 R = A3               % PERFORM SHIFT    07124C00 D
15C2  4809 0000 0B00 00F0          B = B                                     07125C00 D
15C3  2C09 0013 0B00 00F0          IF LC1 THEN B111 = B                      07126C00 D
15C4  4809 0C41 0B00 00FC          B L = B                 % B = 1111/0000 OR B = 0000/0000  07127C00 D
15C5  49C9 EC5C 1E30 C0F0          A3 OR B = A3,B; SET LC1                   07128C00 D
15C6  2000 0000 0000 0060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS  07129C00 D
15C7  4809 E001 2C30 C0F0          A3 L = A2                                 07130C00 D
15C8  0000 0000 0000 002C          COMP 16 = SAR                             07131C00 D
15C9  4809 C0C0 8030 C0FC          A2 R = MIR              % (R(A+1)) SHIFTED INTO MIR  07132C00 D
15CA  2F50 0003 0000 0060          EOUTPUT - 1 = CPCR                        07133C00 D
15CB  4809 0000 0030 20F0          ASR                     % REFERNCE BR1    07134C00 D
15CC  4809 0F40 801C 00F0          BMAR R = MAR2           % R(A)             07135C00 D
15CD  0000 0000 0000 001C          B = SAR                                   07136C00 D
15CE  4809 E0C0 8030 00F0          A3 R = MIR                                07137C00 D
```

```
15CF  3000 0000 0000 0020          16 = SAR                                  07138000 D
15D0  2F5C 0000 0000 C060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A))  07139000 D
15D1  66EC 0000 0000 C040          OPCODE - 1 = MPCR                         07140000 D
                                                          %                  07141000 D
                                                          %                  07142000 D
15D2  5F90 0000 0000 0060  OPCODE61:  XFCODE - 1 = CPCR   % "F" INTO A2      07143000 D
15D3  4809 C640 0030 00F0          A2 + AMPCR = AMPCR                        07144000 D
15D4  1F80 0000 0000 0000          OP61F - 1 = AMPCR                         07145000 D
15D5  4809 0000 0000 00F0          STEP                                      07146000 D
15D6  4824 0000 0000 00F0          EXEC                                      07147000 D
                                                          %                  07148000 D
15D7  1F90 0000 0000 0040  OP61F:  OP610 - 1 = MPCR                          07149000 D
15D8  1FAC 0000 0000 0040          OP611 - 1 = MPCR                          07150000 D
15D9  1FB0 0000 0000 0040          OP612 - 1 = MPCR                          07151000 D
15DA  1FC0 0000 0000 0040          OP613 - 1 = MPCR                          07152000 D
                                                          %                  07153000 D
                                                          %                  07154000 D
                           OP610:                         % RL TYPE ALG LEFT SINGLE SHIFT  07155000 D
                                                          % (R(A)) M (0-3) PLACES, ZERO FILL  07156000 D
                                                          % SET CC AND SET OVERFLOW BIT  07157000 D
15DB  4809 2C56 2030 00F0          LIT AND B = A2         % "M" FIELD INTO A2  07158000 D
15DC  8CF0 0000 0000 0080          15 = LIT; 4 = SAR                         07159000 D
15DD  4809 0C40 8000 00F0          B R = B                                   07160000 D
15DE  4809 2C56 0B00 00F0          LIT AND B = B          % ISOLATE "A"      07161000 D
15DF  51C0 0000 0000 C060          REGSTACK - 1 = CPCR                       07162000 D
15E0  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B    07163000 D
15E1  4809 0002 2030 00F0          NOT A2 = A2                               07164000 D
15E2  4809 0000 0000 80F0          A2 + 1 = SAR                              07165000 D
15E3  4809 0C41 0000 00F0          B L = B                % PERFORM LEFT SHIFT  07166000 D
15E4  4809 0C40 A000 00F0          B R = A2                                  07167000 D
15E5  0000 0000 0000 0020          16 = SAR                                  07168000 D
15E6  4809 0002 0030 00F0          NOT A2                                    07169000 D
15E7  63C9 0000 0000 00F0          IF NOT ABT THEN SET LC1                   07170000 D
15E8  5020 0000 0000 0060          OVBIT - 1 = CPCR       % SET THE OVERFLOW BIT  07171000 D
15E9  4809 0C41 0B30 00F0          B L = B                                   07172000 D
15EA  0000 0000 0000 0020          COMP 16 = SAR                             07173000 D
15EB  4809 0C40 8B30 00F0          B R = MIR, B           % LEFT SHIFTED (R(A))  07174000 E
15EC  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A))  07175000 D
15ED  2000 0000 0000 C060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS  07176000 D
15EE  66E0 0000 0000 C040          OPCODE - 1 = MPCR                         07177000 D
                                                          %                  07178000 D
                                                          %                  07179000 D
                           OP611:                         % TYPE RL CIRCULAR LEFT SHIFT  07180000 D
                                                          % (R(A)) SHIFTED M (0-3) PLACES, SET CC  07181000 D
15EF  4809 2C56 2030 00F0          LIT AND B = A2         % "M" INTO A2  07182000 D
15F0  8CF0 0000 0000 0080          15 = LIT; 4 = SAR                         07183000 D
15F1  4809 0C40 8B00 00F0          B R = B                                   07184000 D
15F2  4809 2C56 0B00 00F0          LIT AND B = B          % ISOLATE "A" FIELD  07185000 D
15F3  51C0 0000 0000 C060          REGSTACK - 1 = CPCR                       07186000 D
15F4  2F30 0000 0030 C060          EINPUT - 1 = CPCR      % (R(A)) INTO B    07187000 D
15F5  4809 0C40 1030 00F0          B = A3                                    07188000 D
15F6  4809 E0C1 1000 00F0          A3 L = A3                                 07189000 D
15F7  0100 0000 0000 00A0          COMP 16 = SAR; 16 = LIT                   07190000 D
15F8  4809 EC5C 1030 00F0          A3 OR B = A3           % A3 = (R(A))/(R(A))  07191000 D
15F9  4809 C000 0B00 00F0          A2 = B                 % "M" INTO B  07192000 D
15FA  4809 2C5E 2000 00F0          LIT - B = A2           % 16 COMPLEMENT OF SHIFT AMOUNT  07193000 D
15FB  4809 C000 0000 80F0          A2 = SAR                                  07194000 D
15FC  4809 E0C1 9030 00F0          A3 C = A3              % PERFORM SHIFT  07195000 D
15FD  4809 E001 1030 00F0          A3 L = A3                                 07196000 D
15FE  0000 0000 0000 0020          COMP 16 = SAR                             07197000 D
```

```
15FF  4809 E000 8030 00F0          A3 R = MIR, B                                  07198000 D
1600  2F50 0000 0040 0060          EOUTPUT - 1 = CPCR     % WRITE OUT NEW (R(A))  07199000 D
1601  2000 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS 07200000 D
1602  56EC 0000 0000 0040          OPCODE - 1 = MPCR                              07201000 D
                                                          %                       07202000 D
                                                          %                       07203000 D
                              OF612:                      % RL TYPE ALG LEFT DOUBLE SHIFT 07204000 D
                                                          % (R(A),R(A+1)), M (0-3) 07205000 D
                                                          % ZERO FILL, SET CC, AND SET OVERFLOW 07206000 D
1603  4809 0C40 8800 00F0          B R = B                                        07207000 D
1604  80F0 0000 0000 0080          4 = SAR; 15 = LIT                              07208000 D
1605  4809 E156 2030 00F0          A3 AND LIT = A2        % "M" INTO A2           07209000 D
1606  4809 2056 0800 00F0          LIT AND B = B          % ISOLATE "A" FIELD     07210000 D
1607  5100 0000 0000 0060          REGSTACK - 1 = CPCR                            07211000 D
1608  4809 0F41 0020 00F0          BMAR L = BR1           % TEMP STORAGE OF R(A)  07212000 D
1609  0000 0000 0000 0030          COMP 8 = SAR                                   07213000 D
160A  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B         07214000 D
160B  4809 0C41 1000 00F0          B L = A3                                       07215000 D
160C  0010 0000 0000 00A0          COMP 16 = SAR ; 1 = LIT                        07216000 D
160D  4809 2F5C 0010 00F0          LIT OR BMAR = MAR2     % R(A+1)                07217000 D
160E  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B       07218000 D
160F  4809 E05C 1000 00F0          A3 OR B = A3           % A3 =(R(A))/(R(A+1))   07219000 D
1610  4809 C002 2000 00F0          NOT A2 = A2                                    07220000 D
1611  4809 C0C0 0000 80F0          A2 + 1 = SAR                                   07221000 D
1612  4809 E000 8800 00F0          A3 R = B                                       07222000 D
1613  4809 0C42 0000 00F0          NOT B                                          07223000 D
1614  6309 0000 0000 00F0          IF NOT ABT THEN SET LC1                        07224000 D
1615  5020 0000 0000 0060          OVBIT - 1 = CPCR       % SET THE OVERFLOW BIT  07225000 D
1616  4809 C0C0 0000 80F0          A2 + 1 = SAR                                   07226000 D
1617  49C9 E001 1800 00F0          A3 L = A3, B; SET LC1                          07227000 D
1618  2000 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS 07228000 D
1619  4809 E001 0800 00F0          A3 L = B                                       07229000 D
161A  0000 0000 0000 0020          COMP 16 = SAR                                  07230000 D
161B  4809 0C40 8090 00F0          B R = MIR                                      07231000 D
161C  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR     % OUTPUT (R(A+1))       07232000 D
161D  4809 00C0 0000 20F0          ASR                    % REFERENCE BR1         07233000 D
161E  4809 0F43 8010 00F0          BMAR R = MAR2          % R(A)                  07234000 D
161F  0000 00C0 0000 0010          B = SAR                                        07235000 D
1620  4809 E000 8090 00F0          A3 R = MIR                                     07236000 D
1621  0000 0000 0000 0020          16 = SAR                                       07237000 D
1622  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR     % OUTPUT (R(A))         07238000 D
1623  56EC 0000 0000 0040          OPCODE - 1 = MPCR                              07239000 D
                                                          %                       07240000 D
                                                          %                       07241000 D
                              OF613:                      % RL TYPE CIRCULAR LEFT DOUBLE SHIFT 07242000 D
                                                          % (R(A),R(A+1)), M (0-3) AND SET CC 07243000 D
1624  4809 2056 2030 00F0          LIT AND B = A2         % "M" INTO A2            07244000 D
1625  80F0 0000 0000 0080          15 = LIT ; 4 = SAR                             07245000 D
1626  4809 0C40 8800 00F0          B R = B                                        07246000 D
1627  4809 2056 0800 00F0          LIT AND B = B          % "A" INTO B            07247000 D
1628  5100 0000 0000 0060          REGSTACK - 1 = CPCR                            07248000 D
1629  4809 0F41 0020 00F0          BMAR L = BR1           % TEMP STORAGE          07249000 D
162A  0000 0000 0000 0030          COMP 8 = SAR                                   07250000 D
162B  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B         07251000 D
162C  4809 0C41 1000 00F0          B L = A3                                       07252000 D
162D  0010 0000 0000 00A0          COMP 16 = SAR; 1 = LIT                         07253000 D
162E  4809 2F5C 0010 00F0          LIT OR BMAR = MAR2     % R(A+1)                07254000 D
162F  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B       07255000 D
1630  4809 E05C 1000 00F0          A3 OR B = A3           % A3 = (R(A))/(R(A+1))  07256000 D
1631  4809 C002 2000 00F0          NOT A2 = A2                                    07257000 D
```

```
1632  4809 0000 0000 80F0                A2 + 1 = SAR                                  07258000 D
1633  49C9 E001 9B00 00F0                A3 C = A3,B; SET LC1   % CIRCULAR SHIFT        07259000 D
1634  2000 0000 0000 006C                SETCCA - 1 = CPCR      % SET THE CONDITION BITS 07260000 D
1635  4809 E001 0B00 00F0                A3 L = B                                      07261000 D
1636  0000 0000 0000 0020                COMP 16 = SAR                                 07262000 D
1637  4809 0C40 8C00 00F0                B R = MIR                                     07263000 D
1638  2F50 0000 0000 0060                EOUTPUT - 1 = CPCR     % OUTPUT (R(A+1))      07264000 D
1639  4809 0000 0000 20F0                ASR                    % REF BR1              07265000 D
163A  4809 0F40 8010 00F0                BMAR R = MAR2                                 07266000 D
163B  0000 0000 0000 0010                B = SAR                                       07267000 D
163C  4809 E000 B000 00FC                A3 R = MIR             % (R(A)) INTO MIR       07268000 D
163D  0000 0000 0000 0020                16 = SAR                                      07269000 D
163E  2F50 0000 0000 0060                EOUTPUT - 1 = CPCR     % OUTPUT SHIFTED (R(A)) 07270000 D
163F  56E0 0000 0000 0040                OPCODE - 1 = MPCR                             07271000 D
                                                                %                      07272000 D
                                                                %                      07273000 D
1640  5F9C 0000 0000 0060     OPCODE62:  XFCODE - 1 = CPCR      % "F" INTO A2           07274000 D
1641  4809 C640 0000 00F0                A2 + AMPCR = AMPCR                            07275000 D
1642  1F0C 0000 0000 00C0                OP62F - 1 = AMPCR                             07276000 D
1643  4809 0000 0000 00FC                STEP                                          07277000 D
1644  4824 0000 0000 00F0                EXEC                                          07278000 D
                                                                %                      07279000 D
1645  1FEC 0000 0000 0040     OP62F:     OP620 - 1 = MPCR                              07280000 D
1646  1FFC 0000 0000 0040                OP621 - 1 = MPCR                              07281000 D
1647  2000 0000 0000 0040                OP622 - 1 = MPCR                              07282000 D
1648  2010 0000 0000 0040                OP623 - 1 = MPCR                              07283000 D
                                                                %                      07284000 D
                                                                %                      07285000 D
                              OP620:                            % TYPE RL SUBTRACT, (R(A))-M=R(A)  07286000 D
                                                                % SET CC, CARRY AND OV BITS        07287000 D
1649  22BC 0000 0000 0060                CONTENTSRA - 1 = CPCR % (R(A)) INTO B                   07288000 D
164A  4809 0C41 2000 00FC                B L = A2               % (R(A)) INTO MS WORD OF A2        07289000 D
164B  00F0 0000 0000 00A0                COMP 16 = SAR; 15 = LIT                               07290000 D
164C  4639 E156 0B00 00F0                A3 AND LIT = B         % "M" INTO B                      07291000 D
164D  4809 0C41 0B00 00F0                B L = B                % "M" INTO MS WORD OF B            07292000 D
164E  4649 CC5E 0000 00F0                A2 - B = MIR; SET LC2                                 07293000 D
164F  7BC9 0000 0000 00F0                IF AOV THEN SET LC1                                   07294000 D
1650  1E70 0000 0000 006C                CARRY - 1 = CPCR                                      07295000 D
1651  4F10 0000 0000 0060                CHECKOV - 1 = CPCR                                    07296000 D
1652  4809 0000 0000 00FC                BMI                                                   07297000 D
1653  4809 0C40 8B00 00F0                B R = MIR,B            % NEW (R(A)) INTO LS WORD OF MIR,B 07298000 D
1654  0000 0000 0000 0020                16 = SAR                                              07299000 D
1655  2000 0000 0000 0060                SETCCA - 1 = CPCR      % SET THE CONDITION BITS 07300000 D
1656  2F50 0000 0000 0060                EOUTPUT - 1 = CPCR     % (R(A))-M INTO R(A)     07301000 D
1657  56E0 0000 0000 0040                OPCODE - 1 = MPCR                             07302000 D
                                                                %                      07303000 D
                                                                %                      07304000 D
                              OP621:                            % TYPE RL LITERAL SUBTRACT DOUBLE  07305000 D
                                                                %(R(A),R(A+1)) - M = R(A),R(A+1)  07306000 D
                                                                % SET CC, CARRY AND OV BITS        07307000 D
1658  4809 2056 0C00 00F0                LIT AND B = MIR        % "M" INTO MIR            07308000 D
1659  80F0 0000 0000 0080                15 = LIT; 4 = SAR                             07309000 D
165A  4809 0C40 8800 00FC                B R = B                % SHIFT "A" TO LS 4 BITS  07310000 D
165B  4809 2055 0B00 00F0                LIT AND B = B          % ISOLATE "A" FIELD      07311000 D
165C  51C0 0000 0000 0060                REGSTACK - 1 = CPCR                           07312000 D
165D  4809 0F41 0020 00F0                BMAR L = BR1           % TEMP STORAGE           07313000 D
165E  0000 0000 0000 0030                COMP 8 = SAR                                  07314000 D
165F  2F30 0000 0000 0060                FINPUT - 1 = CPCR      % (R(A)) INTO B          07315000 D
1660  4809 0C41 2000 00F0                B L = A2               % (R(A)) INTO MS WORD OF A2 07316000 D
1661  001C 0000 0000 00A0                COMP 16 = SAR + 1 = LIT                               07317000 D
```

```
1662  4809 2F5C 0C1C 00F0        LIT OR BMAR = MAR2      % R(A+1)                          07318000 D
1663  2F3C 0000 0000 0060        EINPUT - 1 = CPCR       % (R(A+1)) INTO B                  07319000 D
1664  4809 CC5C 2000 00F0        A2 OR B = A2, BMI       % A2 = (R(A))/(R(A+1))             07320000 D
1665  4849 CC5E 009C 00F0        A2 - 9 = MIR; SET LC2                                     07321000 D
1666  7BC9 0000 0000 00F0        IF AOV THEN SET LC1                                       07322000 D
1667  1E70 0003 003C 0060        CARRY - 1 = CPCR                                          07323000 D
1668  4F1C 0003 0030 0060        CHECKOV - 1 = CPCR                                        07324000 D
1669  49C9 0003 000C 00F0        BMI; SET LC1                                              07325000 D
166A  2000 0003 0030 006C        SETCCA - 1 = CPCR                                         07326000 D
166B  4809 0003 000C 00F0        BMI                                                       07327000 D
166C  4809 0C41 2000 00F0        B L = A2                                                  07328000 D
166D  0000 0003 003C 0020        COMP 16 = SAR                                             07329000 D
166E  4809 0C40 8B30 00F0        B R = B                                                   07330000 D
166F  4809 C0C0 8C3C 00FC        A2 R = MIR              % NEW (R(A+1))                    07331000 D
1670  2F50 00C3 0000 0060        EOUTPUT - 1 = CPCR                                        07332000 D
1671  4809 0003 003C 20F0        ASR                     % REF BR1                         07333000 D
1672  4809 0F43 801C 00F0        BMAR R = MAR2                                             07334000 D
1673  0000 0000 0000 C010        B = SAR                                                   07335000 D
1674  4809 CC40 0030 00F0        B = MIR                 % NEW (R(A))                       07336000 D
1675  2F50 0000 0000 C060        EOUTPUT - 1 = CPCR                                        07337000 D
1676  66E0 0000 0000 C040        OPCODE - 1 = MPCR                                         07338000 D
                                                        %                                 07339000 D
                                                        %                                 07340000 D
                          OP622:                        % TYPE RL LITERAL ADD, (R(A))+M -> R(A)  07341000 D
                                                        % SET CC, CARRY AND OV BITS        07342000 D
1677  22B0 0003 0000 0060        CONTENTSRA - 1 = CPCR % (R(A)) INTO E                     07343000 D
1678  4809 CC41 2030 00F0        B L = A2                                                  07344000 D
1679  0CF0 C000 0030 00A0        COMP 16 = SAR; 15 = LIT                                   07345000 D
167A  4809 E155 0B3C 00F0        A3 AND LIT = B          % ISOLATE "M" FIELD               07346000 D
167B  4809 0C41 0B0C 00F0        B L = B                 % "M" INTO MS WORD OF B           07347000 D
167C  4809 CC40 0090 00F0        A2 + B = MIR            % PERFORM ADDITION                07348000 D
167D  7BC9 00C0 0000 00F0        IF AOV THEN SET LC1                                       07349000 D
167E  1E70 0003 0000 C060        CARRY - 1 = CPCR                                          07350000 D
167F  4F10 0003 0030 0060        CHECKOV - 1 = CPCR                                        07351000 D
1680  4809 00C0 0030 00F0        BMI                                                       07352000 D
1681  4809 0C40 8B9C 00FC        B R = MIR,B                                               07353000 D
1682  0C00 00C0 003C 0020        16 = SAR                                                  07354000 D
1683  2000 00C0 0030 006C        SETCCA - 1 = CPCR       % SET THE CONDITION BITS          07355000 D
1684  2F5C 0000 0000 C060        EOUTPUT - 1 = CPCR      % (R(A))+M INTO R(A)              07356000 D
1685  66EC 0000 C030 0040        OPCODE - 1 = MPCR                                         07357000 D
                                                        %                                 07358000 D
                                                        %                                 07359000 D
                          OP623:                        % TYPE RL LITERAL ADD DOUBLE       07360000 D
                                                        % (R(A),R(A+1)) + M INTO R(A),R(A+1)  07361000 D
                                                        % SET CC, CARRY, AND OV BITS       07362000 D
1686  4809 0C40 8B30 00F0        B R = B                                                   07363000 D
1687  3CFC 00C0 0C30 0080        4 = SAR; 15 = LIT                                         07364000 D
1688  4809 E156 009C 00F0        A3 AND LIT = MIR        % "M" INTO MIR                    07365000 D
1689  4809 2C55 0B3C 00F0        LIT AND B = B           % ISOLATE "A" FIELD               07366000 D
168A  51C0 00C0 0C00 C060        REGSTACK - 1 = CPCR     % R(A) INTO MAR2                  07367000 D
168B  4809 CF81 0C30 00FC        BMAR L = BR1            % TEMP STORAGE                    07368000 D
168C  000C 00C0 0030 C03C        COMP 8 = SAR                                              07369000 D
168D  2F3C 00C0 0000 C06C        EINPUT - 1 = CPCR       % (R(A)) INTO D                   07370000 D
168E  4809 0C41 2030 00F0        B L = A2                % (R(A)) INTO MS WORD OF A2        07371000 D
168F  001C 00C0 003C 00A0        COMP 16 = SAR ; 1 = LIT                                   07372000 D
1690  4809 2F5C 001C 00F0        LIT OR BMAR = MAR2      % R(A+1)                          07373000 D
1691  2F3C 00C0 0000 0060        EINPUT - 1 = CPCR       % (R(A+1)) INTO B                 07374000 D
1692  4809 CC5C 2030 00F0        A2 OR B = A2, BMI       % A2 = (R(A))/(R(A+1))             07375000 D
1693  4809 CC40 0090 00FC        A2 + B = MIR                                              07376000 D
1694  7BC9 0000 0000 00F0        IF AOV THEN SET LC1                                       07377000 D
```

```
1695  1E70 0000 0000 0060                CARRY - 1 = CPCR                                  07378000 D
1696  4F10 0000 0000 0060                CHECKOV - 1 = CPCR                                07379000 D
1697  49C9 C000 0000 00F0                BMI; SET LC1                                      07380000 D
1698  200C 0000 0000 C060                SETCCA - 1 = CPCR                                 07381000 D
1699  4809 C000 0000 C0F0                BMI                                               07382000 D
169A  4809 0041 2000 C0F0                B L = A2                                          07383000 D
169B  0000 0000 0000 0020                COMP 16 = SAR                                     07384000 D
169C  4809 C040 8000 00F0                B R = B                                           07385000 D
169D  4809 C000 8080 00F0                A2 R = MIR                % NEW (R(A+1))          07386000 D
169E  2F50 0000 0000 C060                EOUTPUT - 1 = CPCR                                07387000 D
169F  4809 0000 0000 20F0                ASR                       % REF GR1               07388000 D
16A0  4809 0F40 8010 00FC                BMAR R = MAR2             % R(A)                  07389000 D
16A1  0000 C000 0000 0010                B = SAR                                           07390000 D
16A2  4809 0040 0080 00F0                B = MIR                   % NEW (R(A))            07391000 D
16A3  2F50 0000 0000 C060                EOUTPUT - 1 = CPCR                                07392000 D
16A4  66E0 0000 0000 0040                OPCODE - 1 = MPCR                                 07393000 D
                                                                   %                       07394000 D
                                                                   %                       07395000 D
16A5  5F90 0000 0000 C060        OPCODE63:  XFCODE - 1 = CPCR      % "F" INTO A2            07396000 D
16A6  4809 C040 0040 00F0                A2 + AMPCR = AMPCR                                07397000 D
16A7  2020 0000 0000 0000                OP63F - 1 = AMPCR                                 07398000 D
16A8  4809 0000 0000 C0F0                STEP                                              07399000 D
16A9  4824 0000 0000 00F0                EXEC                                              07400000 D
                                                                   %                       07401000 D
16AA  2030 0000 0000 0040        OP63F:  OP630 - 1 = MPCR                                  07402000 D
16AB  2040 0000 0000 0040                OP631 - 1 = MPCR                                  07403000 D
16AC  2050 0000 0000 0040                OP632 - 1 = MPCR                                  07404000 D
16AD  2060 0000 0000 0040                OP633 - 1 = MPCR                                  07405000 D
                                                                   %                       07406000 D
                                                                   %                       07407000 D
                                 OP630:                            % TYPE RL LITERAL LOAD, M INTO R(A)  07408000 D
                                                                   % SET C*, CARRY AND OV BITS  07409000 D
16AE  4809 2056 0800 C0F0                LIT AND B = B             % ISOLATE "M"            07410000 D
16AF  80F0 0000 0000 0080                15 = LIT; 4 = SAR                                 07411000 D
16B0  4809 0040 0080 00F0                B = MIR                                           07412000 D
16B1  4809 E000 8800 00F0                A3 R = B                                          07413000 D
16B2  4809 2056 0E30 00FC                LIT AND B = B             % ISOLATE "A"           07414000 D
16B3  51C0 0000 0000 0060                REGSTACK - 1 = CPCR                               07415000 D
16B4  2F50 0000 0000 C060                EOUTPUT - 1 = CPCR        % WRITE "M" INTO R(A)   07416000 D
16B5  2180 0000 0000 C060                SET00 - 1 = CPCR                                  07417000 D
16B6  66E0 0000 0000 0040                OPCODE - 1 = MPCR                                 07418000 D
                                                                   %                       07419000 D
                                                                   %                       07420000 D
                                 OP631:                            % RL TYPE LITERAL COMPARE, (R(A)):M  07421000 D
                                                                   % SET CC, CARRY, AND OV BITS  07422000 D
16B7  4809 2056 0080 00F0                LIT AND B = MIR           % "M" INTO MIR           07423000 D
16B8  00F0 0000 0000 00E0                15 = LIT                                          07424000 D
16B9  4809 E000 0800 C0F0                A3 = B                                            07425000 D
16BA  22B0 0000 0000 C060                CONTENTSRA - 1 = CPCR % (R(A)) INTO B             07426000 D
16BB  4809 0041 2000 00F0                B L = A2, BMI             % (R(A)) INTO MS WORD OF A2  07427000 D
16BC  0000 C000 0000 0020                COMP 16 = SAR                                     07428000 D
16BD  4809 0041 0800 00FC                B L = B                   % "M" INTO MS WORD OF B 07429000 D
16BE  2000 0000 0000 0060                SETCC - 1 = CPCR          % SET THE CONDITION BITS 07430000 D
16BF  4849 C05E 0080 C0F0                A2 - B = MIR; SET LC2                             07431000 D
16C0  7BC9 C000 0000 C0F0                IF AOV THEN SET LC1                               07432000 D
16C1  1E70 0000 0000 0060                CARRY - 1 = CPCR                                  07433000 D
16C2  4F10 0000 0000 C060                CHECKOV - 1 = CPCR                                07434000 D
16C3  66E0 0000 0000 0040                OPCODE - 1 = MPCR                                 07435000 D
                                                                   %                       07436000 D
                                                                   %                       07437000 D
```

```
                                OP632:                  % TYPE RL LITERAL MULTIPLY                  07438000 D
                                                        % R(A+1) X M = (R(A),R(A+1)), SET CC        07439000 D
16C4  2809 0C40 8B30 C0FC                B R = B; IF LC1                                            07440000 D
16C5  8CF0 00C0 0030 0080                4 = SAR; 15 = LIT                                          07441000 D
16C6  4809 2C56 0B00 C0F0                LIT AND B = B          % ISOLATE "A"                       07442000 D
16C7  51C0 00C0 0000 C060                REGSTACK - 1 = CPCR                                        07443000 D
16C8  4809 0F41 0030 00F0                BMAR L = BR1           % TEMP STORAGE                      07444000 D
16C9  0000 0000 0030 0030                COMP B = SAR                                               07445000 D
16CA  4809 0F46 0B30 C0FC                BMAR + 1 = B           % R(A+1)                            07446000 D
16CB  51C0 00C0 0030 C060                REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2                  07447000 D
16CC  2F30 00C0 0030 C060                EINPUT - 1 = CPCR      % (R(A+1)) INTO B                   07448000 D
16CD  4809 E156 2000 00F0                A3 AND LIT = A2        % "M" INTO A2                       07449000 D
16CE  00FC 0000 0000 00E0                15 = LIT                                                   07450000 D
16CF  4C60 0000 0030 C060                MULT - 1 = CPCR        % (R(A+1)) X M INTO A3              07451000 D
16D0  4809 E000 0B30 00F0                A3 = B, MIR            % PRODUCT INTO B, MIR               07452000 D
16D1  2000 00C0 C000 C060                SETCCA - 1 = CPCR      % SET THE CONDITION BITS            07453000 D
16D2  4809 E001 0B30 00F0                A3 L = B                                                   07454000 D
16D3  0000 0000 0030 0020                COMP 16 = SAR                                              07455000 D
16D4  4809 0C40 8030 C0F0                B R = MIR              % (R(A+1))                          07456000 D
16D5  2F50 00C0 0030 0060                EOUTPUT - 1 = CPCR                                         07457000 D
16D6  4809 0C00 0C30 20F0                ASR                    % REF BR1                           07458000 D
16D7  4809 0F40 801C 00F0                BMAR R = MAR2          % R(A)                              07459000 D
16D8  0000 00C0 0000 0010                B = SAR                                                    07460000 D
16D9  4809 E000 8030 00F0                A3 R = MIR             % (R(A))                            07461000 D
16DA  0000 0000 0030 0020                16 = SAR                                                   07462000 D
16DB  2F50 00C0 0000 C06C                EOUTPUT - 1 = CPCR                                         07463000 D
16DC  66E0 0000 0030 C040                OPCODE - 1 = MPCR                                          07464000 D
                                                        %                                           07465000 D
                                                        %                                           07466000 D
                                OP633:                  % TYPE RL LITERAL DIVIDE                    07467000 D
                                                        % (R(A),R(A+1))/M = R(A+1), REMAINDER       07468000 D
                                                        % INTO R(A), SET CC AND OV                  07469000 D
16DD  4809 2C56 0B00 00F0                LIT AND B = B          % ISOLATE "M"                       07470000 D
16DE  00FC 00C0 0030 00AC                15 = LIT; COMP 16 = SAR                                    07471000 D
16DF  4809 0C41 0030 00F0                B L = MIR              % LEFT JUSTIFY DIVISOR              07472000 D
16E0  4809 E000 8B00 00FC                A3 R = B                                                   07473000 D
16E1  8000 0000 0000 00C0                4 = SAR                                                    07474000 D
16E2  4809 2C56 0B30 C0F0                LIT AND B = B          % ISOLATE "A"                       07475000 D
16E3  51C0 00C0 0000 C06C                REGSTACK - 1 = CPCR    % R(A) INTO MAR2                    07476000 D
16E4  4809 0F41 0C20 00F0                BMAR L = BR1           % STORE R(A)                        07477000 D
16E5  0000 00C0 0030 0030                COMP B = SAR                                               07478000 D
16E6  2F30 00C0 0000 0060                EINPUT - 1 = CPCR      % (R(A)) INTO B                     07479000 D
16E7  4809 C041 2030 00F0                B L = A2                                                   07480000 D
16E8  0000 00C0 0000 0C20                COMP 16 = SAR                                              07481000 D
16E9  4809 0F46 0B30 C0F0                BMAR + 1 = B           % R(A+1)                            07482000 D
16EA  51C0 00C0 0000 0060                REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2                  07483000 D
16EB  2F30 0000 0000 006C                EINPUT - 1 = CPCR      % (R(A+1)) INTO B                   07484000 D
16EC  49C9 CC50 2D30 00F0                A2 OR B = A2,BMI; SET LC1  % A2 = (R(A))/(R(A+1))          07485000 D
16ED  2400 0000 0000 0060                DIV - 1 = CPCR         % (R(A),R(A+1))/M                   07486000 D
16EE  4809 0000 0030 20F0                ASR                    % REF BR1                           07487000 D
16EF  4809 0F40 801C 00F0                BMAR R = MAR2          % R(A)                              07488000 D
16F0  0000 00C0 0030 0010                B = SAR                                                    07489000 D
16F1  4809 C000 0030 00FC                A2 = MIR               % REMAINDER                         07490000 D
16F2  2F50 00C0 0000 006C                EOUTPUT - 1 = CPCR                                         07491000 D
16F3  4809 E000 0B30 00FC                A3 = B, MIR            % QUOTIENT INTO B, MIR              07492000 D
16F4  2000 0000 0000 C060                SETCCA - 1 = CPCR      % SET THE CONDITION CODE            07493000 D
16F5  4809 0F46 0B30 00F0                BMAR + 1 = B           % R(A+1)                            07494000 D
16F6  51C0 0000 0030 C060                REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2                  07495000 D
16F7  2F50 0000 0030 006C                EOUTPUT - 1 = CPCR     % QUOTIENT INTO R(A+1)              07496000 D
16F8  5050 00C0 0000 C06C                CLEAROV - 1 = CPCR     % SET OV BIT                        07497000 D
```

```
16F9  66EC 0000 0000 0040          OPCODE - 1 = MPCR                                  07498000 D
                                                   %                                  07499000 D
                                                   %                                  07500000 D
16FA  5F90 0000 0000 0060  OPCODE64:  XFCODE - 1 = CPCR   % "F" INTO A2               07501000 D
16FB  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                 07502000 D
16FC  2070 0000 0000 0000          OP64F - 1 = AMPCR                                  07503000 D
16FD  4809 0000 0000 00F0          STEP                                               07504000 D
16FE  4824 0000 0030 00F0          EXEC                                               07505000 D
                                                   %                                  07506000 D
16FF  3000 0000 0000 0040  OP64F:  FAULT - 1 = MPCR                                   07507000 D
1700  3000 0000 0000 0040          FAULT - 1 = MPCR                                   07508000 D
1701  3000 0000 0000 0040          FAULT - 1 = MPCR                                   07509000 D
1702  2080 0000 0000 0040          OP643 - 1 = MPCR                                   07510000 D
                                                   %                                  07511000 D
                                                   %                                  07512000 D
                           OP643:                  % RX TYPE BYTE SUBTRACT             07513000 D
                                                   % (R(A)) - (Y) BYTE INTO R(A)       07514000 D
                                                   % SET CC, CARRY, AND OV BITS        07515000 D
1703  4849 C000 0000 00F0          SET LC2          % BYTE OPERATION FLAG FOR RXMFIELD 07516000 D
1704  5700 0000 0000 0060          RXMFIELD - 1 = CPCR  % Y INTO B                    07517000 D
1705  4809 0041 0010 00F0          B L = BR2                                          07518000 D
1706  0000 0000 0030 0030          COMP 8 = SAR                                       07519000 D
1707  60E0 0000 0000 0060          EMULIN - 1 = CPCR    % (Y) INTO B                  07520000 D
1708  4809 0041 0030 00F0          B L = MIR                                          07521000 D
1709  0000 0000 0030 0020          COMP 16 = SAR                                      07522000 D
170A  4809 E000 0800 00F0          A3 = B                                             07523000 D
170B  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR % (R(A)) INTO E              07524000 D
170C  4809 0041 2030 00F0          B L = A2,BMI         % (R(A)) INTO MS WORD OF A2   07525000 D
170D  0000 0000 0030 0020          COMP 16 = SAR                                      07526000 D
170E  380B 0000 0000 00F0          IF LC2 THEN STEP ELSE SKIP                         07527000 D
170F  2090 0000 0000 0040          LS643 - 1 = MPCR                                   07528000 D
1710  4809 0040 8800 00F0          B R = B              % GET MS BYTE OF (Y)          07529000 D
1711  0000 0000 0000 0030          24 = SAR                                           07530000 D
1712  4809 0041 0800 00F0          B L = B                                            07531000 D
1713  0000 0000 0000 0020          COMP 16 = SAR                                      07532000 D
1714  4849 0055 0080 00F0  C643:   A2 - B = MIR; SET LC2                              07533000 D
1715  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                                07534000 D
1716  1E70 0000 0000 0060          CARRY - 1 = CPCR                                   07535000 D
1717  9F10 0000 0000 0060          CHECKOV - 1 = CPCR                                 07536000 D
1718  4809 0000 0020 00F0          BMI                                                07537000 D
1719  4809 0040 8830 00F0          B R = B, MIR                                       07538000 D
171A  0000 0000 0030 0020          16 = SAR                                           07539000 D
171B  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR   % WRITE NEW (R(A))            07540000 D
171C  2000 0000 0000 0060          SETCCA - 1 = CPCR                                  07541000 D
171D  66EC 0000 0000 0040          OPCODE - 1 = MPCR                                  07542000 D
171E  4809 0041 0800 40F0  LS643:  B L = B, CSAR                                      07543000 D
171F  0000 0000 0000 0030          COMP 8 = SAR                                       07544000 D
1720  4809 0040 8000 00F0          B R = B              % LS BYTE ISOLATED INTO B     07545000 D
1721  7130 0000 0000 005C          C643 - 1  = MPCR                                   07546000 D
                                                   %                                  07547000 D
                                                   %                                  07548000 D
1722  5F90 0000 0000 0060  OPCODE65:  XFCODE - 1 = CPCR   % "F" INTO A2               07549000 D
1723  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                 07550000 D
1724  20A0 0000 0000 0000          OP65F - 1 = AMPCR                                  07551000 D
1725  4809 0000 0000 00F0          STEP                                               07552000 D
1726  4824 0000 0000 00F0          EXEC                                               07553000 D
                                                   %                                  07554000 D
1727  3000 0000 0000 0040  OP65F:  FAULT - 1 = MPCR                                   07555000 D
1728  3000 0000 0000 0040          FAULT - 1 = MPCR                                   07556000 D
1729  3000 0000 0000 0040          FAULT - 1 = MPCR                                   07557000 D
```

```
172A  2080 0000 0000 0040          OP653 - 1 = MPCR
                                                          %                                  07558000 D
                                                          %                                  07559000 D
                                                          %                                  07560000 D
                                   OP653:                 % TYPE RX BYTE ADD                  07561000 D
                                                          % (R(A)) + (Y) 3YTE INTO R(A)      07562000 D
                                                          % SET CC, CARRY, AND OV BITS       07563000 D
172B  4849 0000 0000 00F0          SET LC2                % RXMFIELD BYTE FLAG               07564000 D
172C  5700 0000 0000 0060          RXMFIELD - 1 = CPCR    % Y INTO B                        07565000 D
172D  4809 0C41 0010 00F0          B L = BR2                                                 07566000 D
172E  0000 0000 0000 0030          COMP 8 = SAR                                              07567000 D
172F  50EC 0000 0000 0060          EMULIN - 1 = CPCR                                         07568000 D
1730  4809 0C41 0090 00F0          B L = MIR                                                 07569000 D
1731  0000 0000 0000 0020          COMP 16 = SAR                                             07570000 D
1732  4809 E000 0000 00F0          A3 = B                                                    07571000 D
1733  2280 0000 0000 0060          CONTENTSRA - 1 = CPCR % (R(A)) INTO E                     07572000 D
1734  4809 0C41 2000 00F0          B L = A2, BMI                                             07573000 D
1735  0000 0000 0000 0020          COMP 16 = SAR                                             07574000 D
1736  3808 0000 0000 00F0          IF LC2 THEN STEP ELSE SKIP                                07575000 D
1737  2000 0000 0000 0040          LS653 - 1 = MPCR                                          07576000 D
1738  4809 0C40 8800 00F0          B R = B                % ISOLATE MS BYTE OF (Y)            07577000 D
1739  0000 0000 0000 0030          24 = SAR                                                  07578000 D
173A  4809 0C41 0800 00F0          B L = B                                                   07579000 D
173B  0000 0000 0000 0020          COMP 16 = SAR                                             07580000 D
173C  4809 0C40 0090 00F0          A2 + B = MIR           % PERFORM ADDITION                 07581000 D
173D  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                                       07582000 D
173E  1E70 0000 0000 0060          CARRY - 1 = CPCR                                          07583000 D
173F  3F10 0000 0000 0060          CHECKOV - 1 = CPCR                                        07584000 D
1740  4809 0000 0000 00F0          BMI                                                       07585000 D
1741  4809 0C40 8B90 00F0          B R = B, MIR                                              07586000 D
1742  0000 0000 0000 0020          16 = SAR                                                  07587000 D
1743  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                        07588000 D
1744  2000 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS           07589000 D
1745  56EC 0000 0000 0040          OPCODE - 1 = MPCR                                         07590000 D
1746  4809 0C41 0800 40F0          B L = B, CSAR          % GET LS BYTE OF (Y)               07591000 D
1747  0000 0000 0000 0030          COMP 8 = SAR                                              07592000 D
1748  4809 0C40 8800 00F0          B R = B                                                   07593000 D
1749  7380 0000 0000 0050          C653 - 1 = MPCR                                           07594000 D
                                                          %                                  07595000 D
                                                          %                                  07596000 D
174A  5F90 0000 0000 0060          OPCODE66:  XFCODE - 1 = CPCR                              07597000 D
174B  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                        07598000 D
174C  2000 0000 0000 0000          OP66F - 1 = AMPCR                                         07599000 D
174D  4809 0000 0000 00F0          STEP                                                      07600000 D
174E  4824 0000 0000 00F0          EXEC                                                      07601000 D
                                                          %                                  07602000 D
174F  3000 0000 0000 0040          OP66F:  FAULT - 1 = MPCR                                  07603000 D
1750  3000 0000 0000 0040          FAULT - 1 = MPCR                                          07604000 D
1751  3000 0000 0000 0040          FAULT - 1 = MPCR                                          07605000 D
1752  20E0 0000 0000 0040          OP663 - 1 = MPCR                                          07606000 D
                                                          %                                  07607000 D
                                   OP663:                 % RX TYPE BYTE COMPARE, (R(A)):(Y) BYTE 07608000 D
                                                          % SET CC, CARRY, AND OV BITS       07609000 D
1753  4849 0000 0000 00F0          SET LC2                % BYTE OPERATION FLAG FOR EXMFIELD 07610000 D
1754  5700 0000 0000 0060          RXMFIELD - 1 = CPCR    % Y INTO B                        07611000 D
1755  4809 0C41 0010 00F0          B L = BR2                                                 07612000 D
1756  0000 0000 0000 0030          COMP 8 = SAR                                              07613000 D
1757  50EC 0000 0000 0060          EMULIN - 1 = CPCR      % (Y) INTO B                      07614000 D
1758  4809 0C41 0090 00F0          B L = MIR                                                 07615000 D
1759  0000 0000 0000 0020          COMP 16 = SAR                                             07616000 D
175A  4809 E000 0800 00F0          A3 = B                                                    07617000 D
```

```
175B  228C 0000 0000 0060          CONTENTSRA - 1 = CPCR                          07618000 D
175C  4809 0C41 2D3C 00F0          B L = A2, BMI                                  07619000 D
175D  0000 0000 0000 0020          COMP 16 = SAR                                  07620000 D
175E  380B 0000 0000 00F0          IF LC2 THEN STEP ELSE SKIP                     07621000 D
175F  2CFC 0000 0000 0040          LS663 - 1 = MPCR        % GET LS BYTE OF (Y)   07622000 D
1760  4809 0C40 8600 00F0          B R = B                                        07623000 D
1761  0000 0000 0000 0030          24 = SAR                                       07624000 D
1762  4809 0C41 1B3C 00F0          B L = B, A3             % MS (Y) BYTE INTO MSW OF E, A3  07625000 D
1763  0000 0000 0000 0020          COMP 16 = SAR                                  07626000 D
1764  4849 CC5E 0030 00F0   C663:  A2 - B = MIR; SET LC2                          07627000 D
1765  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                            07628000 D
1766  1E70 0000 0000 0060          CARRY - 1 = CPCR                               07629000 D
1767  4F1C 0000 0000 0060          CHECKOV - 1 = CPCR                             07630000 D
1768  4809 E000 0B00 00F0          A3 = B                  % RESTORE (Y) INTO B   07631000 D
1769  2000 0000 0000 0060          SETCC - 1 = CPCR        % SET THE CONDITION BITS  07632000 D
176A  66E0 0000 0000 0040          OPCODE - 1 = MPCR                              07633000 D
176B  4809 0C41 0B3C 40F0   LS663: B L = B, CSAR                                  07634000 D
176C  0000 0000 0000 C030          COMP 8 = SAR                                   07635000 D
176D  4809 0C40 9B3C 00F0          B R = B, A3                                    07636000 D
176E  763C 0000 0000 0050          C663 - 1 = MPCR                                07637000 D
                                                   %                              07638000 D
                                                   %                              07639000 D
176F  5F90 0000 0000 C060   OPCODF67:  XFCODE - 1 = CFCR                          07640000 D
1770  4809 C640 0C40 00F0          A2 + AMPCR = AMPCR                             07641000 D
1771  2100 0000 0030 C000          OP67F - 1 = AMPCR                              07642000 D
1772  4809 0000 0000 00F0          STEP                                           07643000 D
1773  4824 0000 0030 00F0          EXEC                                           07644000 D
                                                   %                              07645000 D
1774  4E50 0000 0000 0040   OP67F:  NOTIMP - 1 = MPCR                             07646000 D
1775  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                              07647000 D
1776  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                              07648000 D
1777  2110 0000 0030 C040          OP673 - 1 = MPCR                               07649000 D
                                                   %                              07650000 D
                                                   %                              07651000 D
                                   OP673:          % TYPE RX BYTE COMPARE AND INDEX BY 1  07652000 D
                                                   % (R(A)):(Y) BYTE, (R(M))+1 INTO R(M)  07653000 D
                                                   % SET CC, CARRY, AND OV BIT    07654000 D
1778  4849 0000 0000 00F0          SET LC2                 % BYTE OPERATION FLAG FOR RXMFIELD  07655000 D
1779  570C 0000 0000 C060          RXMFIELD - 1 = CFCR     % Y INTO B             07656000 D
177A  4809 0C41 0010 00F0          B L = BR2                                      07657000 D
177B  0000 0000 0000 0030          COMP 8 = SAR                                   07658000 D
177C  60E0 0000 0000 0060          EMULIN - 1 = CPCR       % (Y) INTO B           07659000 D
177D  4809 0C41 0030 00F0          B L = MIR                                      07660000 D
177E  0000 0000 0000 0020          COMP 16 = SAR                                  07661000 D
177F  4809 E000 0B00 00F0          A3 = B                                         07662000 D
1780  228C 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B         07663000 D
1781  4809 0C41 2D3C 00F0          B L = A2, BMI           % (R(A)) INTO MS WORD OF A2  07664000 D
1782  0000 0000 0000 0020          COMP 16 = SAR                                  07665000 D
1783  380B 0000 0000 00F0          IF LC2 THEN STEP ELSE SKIP                     07666000 D
1784  212C 0000 0000 004C          LS673 - 1 = MPCR                               07667000 D
1785  4809 0C40 9B00 00F0          B R = B                                        07668000 D
1786  0000 0000 0000 0030          24 = SAR                                       07669000 D
1787  4809 0C41 1B3C 00F0          B L = B, A3             % MS  (Y) BYTE INTO MS WORD OF B  07670000 D
1788  0000 0000 0000 0020          COMP 16 = SAR                                  07671000 D
1789  4849 CC5E 0030 00F0   C673:  A2 - B = MIR; SET LC2                          07672000 D
178A  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                            07673000 D
178B  1E70 0000 0000 0060          CARRY - 1 = CPCR                               07674000 D
178C  4F1C 0000 0000 0060          CHECKOV - 1 = CPCR                             07675000 D
178D  4809 E000 0B00 00F0          A3 = B                                         07676000 D
178E  2000 0000 0000 C060          SETCC - 1 = CPCR                               07677000 D
```

```
178F  2380 6000 0000 0060          CONTENTSRM - 1 = CPCR   % (R(P)) INTO B          07678000 D
1790  4809 0046 0030 00F0          B + 1 = MIR             % (R(M)) + 1              07679000 D
1791  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR                               07680000 D
1792  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                07681000 D
1793  4809 0041 0800 40F0   LS673:  B L = B, CSAR           % ISOLATE LS BYTE OF (Y) 07682000 D
1794  000C 0000 0000 0030          COMP B = SAR                                     07683000 D
1795  4809 0040 9B30 00F0          B R = B, A3                                      07684000 D
1796  788C 0000 0000 0050          C673 - 1 = MPCR                                  07685000 D
                                                        %                          07686000 D
                                                        %                          07687000 D
                              OPCODE70:                 % NOT IMPLEMENTED          07688000 D
1797  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                                07689000 D
                                                        %                          07690000 D
                              OPCODE71:                 % NOT IMPLEMENTED          07691000 D
1798  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                                07692000 D
                                                        %                          07693000 D
                              OPCODE72:                 % NOT IMPLEMENTED          07694000 D
1799  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                                07695000 D
                                                        %                          07696000 D
                              OPCODE73:                 % NOT IMPLEMENTED          07697000 D
179A  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                                07698000 D
                                                        %                          07699000 D
                              OPCODE74:                 % NOT IMPLEMENTED          07700000 D
179B  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                                07701000 D
                                                        %                          07702000 D
                              OPCODE75:                 % NOT IMPLEMENTED          07703000 D
179C  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                                07704000 D
                                                        %                          07705000 D
                              OPCODE76:                 % NOT IMPLEMENTED          07706000 D
179D  4E50 0000 0000 0040          NOTIMP - 1 = MPCR                                07707000 D
                                                        %                          07708000 D
                              OPCODE77:                 % NOT ASSIGNED             07709000 D
179E  3000 0000 0000 0040          FAULT - 1 = MPCR                                 07710000 D
                                                        %                          07711000 D
179F  0000 0000 0000 0040          END                                              07712000 D
1784  792C 0000 0000 0050   IEND
1777  777C 0000 0000 0050
1771  773C 0000 0000 0000
175F  76AC 0000 0000 0050
1752  7520 0000 0000 0050
174C  74E0 0000 0000 0000
1737  7450 0000 0000 0050
172A  72A0 0000 0000 0050
1724  7260 0000 0000 0000
170F  710C 0000 0000 0050
1702  702C 0000 0000 0050
16FC  6FE0 0000 0000 0000
16AD  5DCC 0000 0000 0050
16AC  6C30 0000 0000 0050
16AB  5B60 0000 0000 0050
16AA  6A0C 0000 0000 0050
16A7  6A90 0000 0000 0000
1648  6850 0000 0000 0050
1647  576C 0000 0000 0050
1646  6570 0000 0000 0050
1645  648C 0000 0000 0050
1642  644C 0000 0000 0000
150A  623C 0000 0000 0050
1509  6020 0000 0000 0050
150B  5EEC 0000 0000 0050
```

1507	5DA0	0000	0000	005C
1504	506C	0000	0000	0000
1573	5AAC	0000	0030	0050
1572	58FC	0000	0000	C050
1571	57B0	0000	C00C	C050
1570	573C	0000	0000	0050
156D	56FC	0000	0000	00DC
152D	552C	C000	0000	C050
152B	53E0	0000	0000	C050
152A	520C	0000	0000	0050
1527	529C	0000	0000	0000
14F5	50FC	0000	0000	C050
14F3	506C	0000	0000	0050
14F2	4F5C	0000	0000	C05C
14EF	4F1C	C000	0000	0000
14B2	4D4C	0000	0000	0050
14B1	4C50	0000	C000	C050
14B0	4BE0	0000	0000	005C
14AF	4B20	0000	0000	C050
14AC	4AE0	0000	0000	C000
1470	495C	0000	0000	0050
146F	4820	0000	0000	C050
146E	47C0	0000	0000	C050
146D	4700	0000	0000	C050
146A	4600	0000	0000	0000
1436	456C	0000	0000	005C
1435	4450	0000	0000	C05C
1434	4400	0000	0000	0050
1433	436C	0000	0000	005C
1430	4320	0000	0000	C00C
13FF	41FC	0000	0000	C050
13FE	40EC	0000	0000	0050
13FD	4090	0000	0000	0050
13FC	3FFC	0000	0000	005C
13F9	3FBC	0000	0000	0000
13B2	3E60	0000	0000	0050
13B1	3D00	0000	0030	0050
13B0	3B2C	0000	0000	0050
13AC	3AE0	0000	0000	0000
1371	396C	0000	0000	C050
1370	3800	0000	0000	C050
136E	371C	0000	0000	0050
136B	3600	0000	0000	C000
1328	353C	0000	0030	C050
132A	3300	0000	0000	005C
1329	338C	0000	0000	0050
1328	32B0	0000	C000	C050
1325	3270	0000	0000	C00C
12FA	3180	0000	0000	0050
12F9	305C	0000	0000	005C
12F7	2FAC	0000	0000	005C
12F4	2F6C	0000	0000	C00C
12F1	2F1C	0000	0000	C05C
12EE	2F1C	0000	0000	C050
12E8	2F1C	0000	0000	C05C
12E5	2F1C	0000	0000	C050
12E0	2F1C	0000	0000	C050
12DD	2F1C	0000	0000	0050
12DC	2F1C	0000	0000	0050
12D7	2F1C	0000	0030	C05C

12D2	2F1C	0000	0000	0000	0050
12CD	2F10	0000	0000	0000	0050
12C8	2F10	0000	0000	0000	0050
12C4	2F1C	0000	0000	0000	0050
12C0	2E8C	0000	0000	0000	0050
12BF	2E00	0000	0000	0000	0050
12BF	2DDC	0000	0000	0000	0050
12BD	2DCC	0000	0000	0000	0050
12BA	2D7C	0000	0000	0000	0050
12B9	2D2C	0000	0000	0000	0050
12B8	2CDC	0000	0000	0000	0050
12B7	2C8C	0000	0000	0000	0050
12B6	2C4C	0000	0000	0000	0050
12B5	2C00	0000	0000	0000	0050
12B2	2B40	0000	0000	0000	0000
12A9	321C	0000	0000	0000	0050
1291	2990	0000	0000	0000	0050
1264	26D0	0000	0000	0000	0050
1244	28BC	0000	0000	0000	0050
1243	27BC	0000	0000	0000	0050
1242	25B0	0000	0000	0000	0050
1241	2440	0000	0000	0000	0050
123E	2400	0000	0000	0000	0000
11EB	228C	0000	0000	0000	0050
11EA	20EC	0000	0000	0000	0050
11E9	1FBC	0000	0000	0000	0050
11E8	1EBC	0000	0000	0000	0050
11E5	1E7C	0000	0000	0000	0000
117A	1C9C	0000	0000	0000	0050
1179	1A9C	0000	0000	0000	0050
1178	1900	0000	0000	0000	0050
1177	17A0	0000	0000	0000	0050
1174	176C	0000	0000	0000	0000
113C	1650	0000	0000	0000	0050
113B	152C	0000	0000	0000	0050
113A	146C	0000	0000	0000	0050
1139	13C0	0000	0000	0000	0050
1136	1380	0000	0000	0000	0000
10FE	127C	0000	0000	0000	0050
10FD	1140	0000	0000	0000	0050
10FC	108C	0000	0000	0000	0050
10FB	0FEC	0000	0000	0000	0050
10F8	0FAC	0000	0000	0000	0000
10C0	0E9C	0000	0000	0000	0050
10BF	0D60	0000	0000	0000	0050
10BE	0CAC	0000	0000	0000	0050
10BD	0C0C	0000	0000	0000	0050
10BA	0BCC	0000	0000	0000	0000
1017	092C	0000	0000	0000	0050
1016	063C	0000	0000	0000	0050
1015	03CC	0000	0000	0000	0050
1014	017C	0000	0000	0000	0050
1011	013C	0000	0000	0000	0000
0F8C	FFCC	0000	0000	0000	0040
0F8B	FCB0	0000	0000	0000	0040
0F8A	FAAC	0000	0000	0000	0040
0F89	F8CC	0000	0000	0000	0040
0F86	F88C	0000	0000	0000	0000
0F30	F68C	0000	0000	0000	0040
0F2E	F48C	0000	0000	0000	0040

0F2D	F300	0000	0000	0000	0040
0F2A	F2C0	0000	0000	0000	0000
0FDE	F150	0000	0000	0000	0040
0FDD	EFD0	0000	0000	0000	0040
0EDC	EEC0	0000	0000	0000	0040
0FDB	EDE0	0000	0000	0000	0040
0FD8	EDA0	0000	0000	0000	0000
0E5E	EAD0	0000	0000	0000	0040
0F5C	E850	0000	0000	0000	0040
0E5B	E5E0	0000	0000	0000	0040
0E58	E5A0	0000	0000	0000	0000
0F00	E410	0000	0000	0000	0040
0DFF	E270	0000	0000	0000	0040
0DFE	E120	0000	0000	0000	0040
0DFD	E00C	0000	0000	0000	0040
0DFA	DFC0	0000	0000	0000	0000
0D79	DCDC	0000	0000	0000	0040
0D77	DA1C	0000	0000	0000	0040
0D76	D79C	0000	0000	0000	0040
0D73	D75C	0000	0000	0000	0000
0D19	D5C0	0000	0000	0000	0040
0D18	D42C	0000	0000	0000	0040
0D17	D2B0	0000	0000	0000	0040
0D16	D190	0000	0000	0000	0040
0D13	D150	0000	0000	0000	0000
0CBF	D0A0	0000	0000	0000	0040
0CBE	CE30	0000	0000	0000	0040
0CBD	CDD0	0000	0000	0000	0040
0CBC	CBF0	0000	0000	0000	0040
0CB9	CBB0	0000	0000	0000	0000
0C32	C900	0000	0000	0000	0040
0C31	C710	0000	0000	0000	0040
0C30	C560	0000	0000	0000	0040
0C2F	C320	0000	0000	0000	0040
0C2C	C2E0	0000	0000	0000	0000
0BD5	C1D0	0000	0000	0000	0040
0BD4	BF90	0000	0000	0000	0040
0BD3	BEB0	0000	0000	0000	0040
0BD2	BD50	0000	0000	0000	0040
0BCF	BD10	0000	0000	0000	0000
0BBC	BC60	0000	0000	0000	0060
0B97	BA30	0000	0000	0000	0040
0B6F	B7B0	0000	0000	0000	0040
0B64	BAA0	0000	0000	0000	0040
0B63	B820	0000	0000	0000	0040
0B61	B640	0000	0000	0000	0040
0B5E	B600	0000	0000	0000	0000
0AE9	B380	0000	0000	0000	0040
0AE8	B000	0000	0000	0000	0040
0AE6	AE90	0000	0000	0000	0040
0AE3	AE50	0000	0000	0000	0000
0A72	ACA0	0000	0000	0000	0040
0A71	AA30	0000	0000	0000	0040
0A70	A910	0000	0000	0000	0040
0A6F	A720	0000	0000	0000	0040
0A6C	A6E0	0000	0000	0000	0000
0A1F	A5D0	0000	0000	0000	0040
0A1E	A3D0	0000	0000	0000	0040
0A1D	A360	0000	0000	0000	0040
0A1C	A1F0	0000	0000	0000	0040

0A19	A19C	0000	0000	0000	
0A0D	A120	0000	0000	0040	
0A05	A00C	0000	0000	0040	
0900	9F8C	0000	0000	0040	
090C	9E7C	0000	0000	0040	
090A	9D0C	0000	0000	0040	
0907	9D9C	0000	0000	0000	
0998	99EC	0000	0000	0040	
098D	9D2C	0000	0000	0040	
0989	9A4C	0000	0000	0040	
0938	98B0	0000	0000	0040	
0935	987C	0000	0000	0000	
0962	97FC	0000	0000	0040	
0960	970C	0000	0000	0040	
095F	9620	0000	0000	0040	
095C	95EC	0000	0000	0000	
0934	957C	0000	0000	0040	
0932	9420	0000	0000	0040	
0931	934C	0000	0000	0040	
092E	93CC	0000	0000	0000	
0908	90FC	0000	0000	0040	
0909	90BC	0000	0000	0040	
0900	9CFC	0000	0000	0040	
08FE	90F0	0000	0000	0040	
08A7	8ABC	0000	0000	0040	
08BE	8ECC	0000	0000	0040	
08BD	8DBC	0000	0000	0040	
08BC	9C8C	0000	0000	0040	
08B8	88EC	0000	0000	0040	
0888	88AC	0000	0000	0000	
0890	916C	0000	0000	0040	
0b7D	9800	0000	0000	0040	
087A	97CC	0000	0000	0000	
07FF	842C	0000	0000	0040	
07FE	82CC	0000	0000	0040	
07FD	822C	0000	0000	0040	
07FB	812C	0000	0000	0040	
07FA	809C	0000	0000	0040	
07F9	7FFC	0000	0000	0040	
07F3	7F80	0000	0000	0000	
07EF	951C	0000	0000	0040	
07EC	7EFC	0000	0000	0040	
07E9	7E80	0000	0000	0000	
0771	777C	0000	0000	0040	
0765	771C	0000	0000	0040	
073E	700C	0000	0000	0040	
073D	7C3C	0000	0000	0040	
073C	7B60	0000	0000	0040	
073B	7A9C	0000	0000	0040	
0739	7A3C	0000	0000	0040	
0738	78B0	0000	0000	0040	
0737	77F0	0000	0000	0040	
0735	75D0	0000	0000	0040	
0734	74F0	0000	0000	0040	
0733	73EC	0000	0000	0040	
072D	7320	0000	0000	0000	
0724	7E3C	0000	0000	0040	
0722	7D0C	0000	0000	0040	
0721	7240	0000	0000	0040	
071E	720C	0000	0000	0000	

06D0	7CD0	0000	0000	0000	0040
06DC	6F80	0000	0000	0000	0040
06D8	6E90	0000	0000	0000	0040
06DA	6D00	0000	0000	0000	0040
06D7	5D90	0000	0000	0000	0000
06C4	6C40	0000	0000	0000	0040
06BE	6C00	0000	0000	0000	0000
06B8	79D0	0000	0000	0000	0050
06BA	79C0	0000	0000	0000	0050
06B9	79B0	0000	0000	0000	0050
06B8	79A0	0000	0000	0000	0050
06B7	7990	0000	0000	0000	0050
06B6	7980	0000	0000	0000	0050
06B5	7970	0000	0000	0000	0050
06B4	7960	0000	0000	0000	0050
06B3	76E0	0000	0000	0000	0050
06B2	7490	0000	0000	0000	0050
06B1	7210	0000	0000	0000	0050
06B0	6F90	0000	0000	0000	0050
06AF	6A40	0000	0000	0000	0050
06AE	63F0	0000	0000	0000	0050
06AD	5D10	0000	0000	0000	0050
06AC	56A0	0000	0000	0000	0050
06AB	5690	0000	0000	0000	0050
06AA	5680	0000	0000	0000	0050
06A9	5240	0000	0000	0000	0050
06A8	4EC0	0000	0000	0000	0050
06A7	4EB0	0000	0000	0000	0050
06A6	4EA0	0000	0000	0000	0050
06A5	4E90	0000	0000	0000	0050
06A4	4E80	0000	0000	0000	0050
06A3	4A90	0000	0000	0000	0050
06A2	4670	0000	0000	0000	0050
06A1	4200	0000	0000	0000	0050
06A0	3F60	0000	0000	0000	0050
069F	3A90	0000	0000	0000	0050
069E	3680	0000	0000	0000	0050
069D	3220	0000	0000	0000	0050
069C	2A50	0000	0000	0000	0050
069B	2A40	0000	0000	0000	0050
069A	2A30	0000	0000	0000	0050
0699	2380	0000	0000	0000	0050
0698	1E20	0000	0000	0000	0050
0697	1710	0000	0000	0000	0050
0696	1330	0000	0000	0000	0050
0695	0F50	0000	0000	0000	0050
0694	0B70	0000	0000	0000	0050
0693	00E0	0000	0000	0000	0050
0692	F830	0000	0000	0000	0040
0691	F270	0000	0000	0000	0040
0690	ED50	0000	0000	0000	0040
068F	E550	0000	0000	0000	0040
068E	DF70	0000	0000	0000	0040
068D	D700	0000	0000	0000	0040
068C	D100	0000	0000	0000	0040
068B	CB60	0000	0000	0000	0040
068A	C290	0000	0000	0000	0040
0689	BCC0	0000	0000	0000	0040
0688	B5B0	0000	0000	0000	0040
0687	AED0	0000	0000	0000	0040

0686	A690	0000	0000	0000	0040
0685	A160	0000	0000	0000	0040
0684	7040	0000	0000	0000	0040
0683	9820	0000	0000	0000	0040
0682	9590	0000	0000	0000	0040
0681	92B0	0000	0000	0000	0040
0680	8770	0000	0000	0000	0040
067F	7E60	0000	0000	0000	0040
067E	7190	0000	0000	0000	0040
067D	6040	0000	0000	0000	0040
067C	6BB0	0000	0000	0000	0040
0679	67B0	0000	0000	0000	0000
0626	64F0	0000	0000	0000	0040
0613	64F0	0000	0000	0000	0040
060E	61B0	0000	0000	0000	0040
0600	62B0	0000	0000	0000	0040
05D6	66E0	0000	0000	0000	0040
0595	5B50	0000	0000	0000	0040
05B0	5B50	0000	0000	0000	0040
05AB	5B50	0000	0000	0000	0040
05A6	5B50	0000	0000	0000	0040
05A1	5B00	0000	0000	0000	0040
05A0	5AB0	0000	0000	0000	0040
059D	5A60	0000	0000	0000	0040
059A	5A10	0000	0000	0000	0040
0587	6330	0000	0000	0000	0060
0582	6330	0000	0000	0000	0060
0581	5860	0000	0000	0000	0040
0580	5930	0000	0000	0000	0040
0570	5860	0000	0000	0000	0040
0579	5810	0000	0000	0000	0040
0567	56E0	0000	0000	0000	0040
0550	5330	0000	0000	0000	0060
0522	5270	0000	0000	0000	0040
051C	5D60	0000	0000	0000	0040
051B	5300	0000	0000	0000	0060
051A	5E50	0000	0000	0000	0060
0518	5E50	0000	0000	0000	0060
0511	5E50	0000	0000	0000	0060
0505	5050	0000	0000	0000	0040
0504	5CA0	0000	0000	0000	0040
0502	50A0	0000	0000	0000	0040
0501	5050	0000	0000	0000	0040
04FF	5050	0000	0000	0000	0040
04FE	5CA0	0000	0000	0000	0040
04FC	4FF0	0000	0000	0000	0040
04F8	5050	0000	0000	0000	0040
04F1	50F0	0000	0000	0000	0040
04EC	5B00	0000	0000	0000	0060
04E0	50F0	0000	0000	0000	0040
04E6	5B00	0000	0000	0000	0060
04D3	4E20	0000	0000	0000	0040
04CA	4DF0	0000	0000	0000	0040
04AF	61E0	0000	0000	0000	0060
04A6	50E0	0000	0000	0000	0060
0496	51C0	0000	0000	0000	0060
048F	60E0	0000	0000	0000	0060
048A	60E0	0000	0000	0000	0060
047D	4780	0000	0000	0000	0040
0475	60EC	0000	0000	0000	0060

046F	46F0	0000	0000	0000	0040
0460	51C0	0000	0000	0000	0060
0469	46FC	0000	0000	0000	0040
0467	51CC	0000	0000	0000	0060
0464	46FC	0000	0000	0000	0040
0462	51C0	0000	0000	0000	0060
045F	47B0	0000	0000	0000	0040
045D	50EC	0000	0000	0000	0060
0458	469C	0000	0000	0000	0040
0457	464C	0000	0000	0000	0040
0456	45F0	0000	0000	0000	0040
0455	4580	0000	0000	0000	0040
0452	454C	0000	0000	0000	0000
044A	50E0	0000	0000	0000	0060
0444	444C	0000	0000	0000	0040
0442	51C0	0000	0000	0000	0060
043E	4440	0000	0000	0000	0040
043C	444C	0000	0000	0000	0040
043A	51C0	0000	0000	0000	0060
0436	60EC	0000	0000	0000	0060
0431	43E0	0000	0000	0000	0040
0430	43C0	0000	0000	0000	0040
042F	437C	0000	0000	0000	0040
042E	4310	0000	0000	0000	0040
042B	4200	0000	0000	0000	0000
0425	44C0	0000	0000	0000	0040
0424	4250	0000	0000	0000	0040
041F	60E0	0000	0000	0000	0060
0416	416C	0000	0000	0000	0040
0415	6330	0000	0000	0000	0060
0410	416C	0000	0000	0000	0040
0406	6330	0000	0000	0000	0060
03F0	66E0	0000	0000	0000	0040
03E3	3F70	0000	0000	0000	0060
03D5	66E0	0000	0000	0000	0040
03C8	3F70	0000	0000	0000	0060
03BE	4E50	0000	0000	0000	0040
03BD	3D00	0000	0000	0000	0040
03BC	3BE0	0000	0000	0000	0040
03B9	3BB0	0000	0000	0000	0000
03AC	66E0	0000	0000	0000	0040
03A2	3F70	0000	0000	0000	0060
03AE	66E0	0000	0000	0000	0040
0383	3F70	0000	0000	0000	0060
0376	4E50	0000	0000	0000	0040
0375	399C	0000	0000	0000	0040
0374	376C	0000	0000	0000	0040
0371	3730	0000	0000	0000	0000
035E	3620	0000	0000	0000	0040
0357	51E0	0000	0000	0000	0060
0350	50E0	0000	0000	0000	0060
0348	61E0	0000	0000	0000	0060
0340	60E0	0000	0000	0000	0060
0310	61E0	0000	0000	0000	0060
0314	60E0	0000	0000	0000	0060
030F	60E0	0000	0000	0000	0060
0305	50F0	0000	0000	0000	0040
0301	5BD0	0000	0000	0000	0060
02F0	2F30	0000	0000	0000	0060
02E0	2F30	0000	0000	0000	0060

02E9	2F3C	0000	0000	0060
02E6	2F3C	0000	0000	0060
02E3	2F3C	0000	0000	0060
02E0	2F3C	0000	0000	0060
02DD	5E5C	0000	0000	0060
02DC	5E5C	0000	0000	0060
02DB	5E5C	0000	0000	0060
02DA	5E5C	0000	0000	0060
02D8	5E5C	0000	0000	0060
02CE	5E5C	0000	0000	0060
02CB	5E5C	0000	0000	0060
02C8	5E5C	0000	0000	0060
02C5	5E5C	0000	0000	0060
02C2	5E5C	0000	0000	0060
02BA	2F5C	0000	0000	0060
02B1	2F5C	0000	0000	0060
029C	2F3C	0000	0000	0060
0294	2F5C	0000	0000	0060
028F	2F5C	0000	0000	0060
028C	2F5C	0000	0000	0060
0289	2F5C	0000	0000	0060
0286	2F5C	0000	0000	0060
0283	2F5C	0000	0000	0060
0261	266C	0000	0000	0040
025E	261C	0000	0000	0040
0255	27CC	0000	0000	0040
024D	4EBC	0000	0000	0040
023D	2F3C	0000	0000	0060
023C	51C0	0000	0000	0060
0236	2F3C	0000	0000	0060
0230	51C0	0000	0000	0060
0216	216C	0000	0000	0040
0215	21F0	0000	0000	0040
0213	21B0	0000	0000	0040
020D	216C	0000	0000	0040
020C	21F0	0000	0000	0040
0209	21F0	0000	0000	0040
0208	216C	0000	0000	0040
0205	2090	0000	0000	0040
0203	219C	0000	0000	0040
01FB	2F5C	0000	0000	0060
01EA	1EA0	0000	0000	0040
01E9	1EF0	0000	0000	0040
01E1	66E0	0000	0000	0040
01DA	66E0	0000	0000	0040
01B5	1BA0	0000	0000	0040
019F	1A4C	0000	0000	0040
0161	27E0	0000	0000	0060
013C	144C	0000	0000	0060
0124	169C	0000	0000	0040
010C	144C	0000	0000	0060
00F7	0FB0	0000	0000	0040
00E2	0FB0	0000	0000	0040
00ED	0F7C	0000	0000	0040
00EC	0F2C	0000	0000	0040
00E9	0ED0	0000	0000	0040
0085	0B70	0000	0000	0040
0092	0960	0000	0000	0040
0077	0790	0000	0000	0060
006E	0790	0000	0000	0060

```
0066  0830 0000 0000 0060
0066  0810 0000 0000 0060
0058  0970 0000 0000 0060
005A  0870 0000 0000 0060
0059  05B0 0000 0000 0040
0051  5CFC 0000 0000 0040
0050  0810 0000 0000 0060
0049  0510 0000 0000 0060
0046  05EC 0000 0000 0040
0042  0510 0000 0000 0060
0041  0460 0000 0000 0060
0036  0600 0000 0000 0040
0035  1540 0000 0000 0040
0032  1000 0000 0000 0040
002F  1810 0000 0000 0040
002C  1BFC 0000 0000 0040
0029  1AAC 0000 0000 0040
0026  1620 0000 0000 0040
0023  1790 0000 0000 0040
0020  0360 0000 0000 0040
001D  0BF0 0000 0000 0040
001A  0480 0000 0000 0040
0014  0A80 0000 0000 0040
0011  0000 0000 0000 0040
000A  0460 0000 0000 0060
0007  0600 0000 0000 0040
 0 ERRORS    7714 CARDS.  36055 TOKENS.     65156 RULES.      5275 SECONDS.
 0 WARNINGS.      272 DISK SECTORS.
```

BIBLIOGRAPHY

1. Air Force Avionics Lab, Wright-Patterson AFB, Ohio, AFAL-TR-74-180, A Stack Machine Emulation, Anderson, C. M., January 1975.

2. Agrawala, A. K. and Rausher, T. G., "Microprogramming: Perspective and Status," IEEE Trans, C23, p. 817-37, August 1974.

3. Burkhard, W. A., "Flexible Computer Architectures for Research and Development," Computer Conference, Fall 1976.

4. Burroughs Corporation Report 64116, Emulation Facility, by Advanced Design Organization, Paoli, Pa., 28 June 1971.

5. Burroughs Corporation Report TR70-2, The Interpreter, by R. L. Davis, 16 February 1970.

6. Burroughs Corporation Report TR70-8, Microprogramming Manual for the Interpreter Based Systems, by Advanced Design Organization, Paoli, Pa., November 1970.

7. Burroughs Corporation Report 66143, Algol Reference Manual for the Interpreter Based System, by Advanced Design Organization, Paoli, Pa., 15 June 1975.

8. Computer Sciences Corporation, A Cost-Effective Emulation Approach for U. S. Army Telecommunications, by J. W. Carroll, p. 1-13.

9. Naval Electronics Laboratory Center and M. I. T. Sloan School of Management, Facilities Orientation Report, Volume 1, A Survey of the Navy Tactical Computer Applications and Executives, by Professors S. E. Madnick and J. D. Detreville, October 1975.

10. Haggerty, J. M. and Hartling, J. M., Emulation of the AN/UYK-7 Tactical Data Computer on the Burroughs D-Machine, Masters Thesis, Naval Postgraduate School, Monterey, California, December 1976.

11. Husson, S. S., Microprogramming: Principles and Practices, Prentice-Hall, Inc., 1970.

12. Jones, L., "Instruction Sequencing in Microprogrammed Computers," Proceedings NCC, 44, p. 91-8, 1975.

13. Lichstein, H. A., "When Should You Emulate?," Datamation, 15, p. 205-10, November 1969.

14. Mallach, E. G., "Emulator Architecture," Computer, p. 24-32, August 1975.

15. Mercer, R. J., "Microprogramming," Journal for the Association of Computing Machines, 4, p. 157-71, April 1957.

16. Naval Material Command UNCLASSIFIED Letter MAT 09Y:JER Serial 130 to Naval Electronic Systems Command, Subject: Standard Shipboard Tactical Digital Processors and Program Languages, 29 May 1973.

17. Reigel, E. W., Faber, U., and Fisher, D. A., "The interpreter - A microprogramming building block system," Proceedings SJCC, 40, p. 705-23, 1972.

18. Rosin, R. F., "Contempory Concepts of Microprogramming and Emulation," Computer Surveys, 1, p. 197-212, December 1969.

19. Saal, H. J. and Schuster, L. J., On Measuring Computer Systems by Microprogramming.

20. Sperry Rand, UNIVAC, Computer Set AN/UYK20 Technical Manual Operations and Maintenance with Parts List, N 00039-73-C-0432, September 1974.

21. Sperry Rand, UNIVAC, AN/UYK-20 Computer Repertiore of Instructions, Fall 1976.

22. Sperry Rand, UNIVAC, AN/UYK-20 Technical Description, November 1976.

23. Sperry Rand, UNIVAC, User's Handbook for AN/UYK-20(V) Computer Volume 3 (Change 4), NAVELEX 0967-LP-598-2030, April 1976.

24. Wilkes, M. B., "The Growth of Interest in Microprogramming: A Literature Survey," Computer Surveys, 1, p. 139-45, September 1969.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexanderia, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
4.	LT Lyle V. Rich, SC, USN, Code 52Rs (Thesis Advisor) Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
5.	Asst Professor V. Michael Powers, Code 52Pw (Second Reader) Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
6.	Chief of Naval Material (Attn: Code 09Y) Department of the Navy Washington, D. C. 20360	1
7.	Mr. J. Lynch Burroughs ADO Federal and Special Systems Group P. O. Box 517 Paoli, Pennsylvania 19301	1
8.	Mr. C. Benson Naval Electronics Systems Engineering Center P. O. Box 37 San Diego, California 92138	1

9. Mr. J. Lopata 1
Burroughs Corporation
P. O. Box 517
Paoli, Pennsylvania 19301

10. Mr. J. Westergren 1
Field Engineer
Univac Computer Systems
P. O. Box 3525
St. Paul, Minnesota 55165

11. CAPT Ralph H. Anzelmo, USMC 1
15767 Edgewood Drive
Montclair Country Club Lake
Dumfries, Virginia 22026

12. LT Theodore L. Kaye, USN 1
3880 Fairview Road
Reno, Nevada 89511

6 SEP 78 25394
28 FEB 81 26385

Thesis 169582
A567 Anzelmo
c.1 Emulation of the
AN/UYK-20 tactical data
computer on the Bur-
roughs D-machine.
6 SEP 78 25394
28 FEB 81 26385

Thesis 169582
A567 Anzelmo
c.1 Emulation of the
AN/UYK-20 tactical data
computer on the Bur-
roughs D-machine.

BOUND BY HNM

thesA567
Emulation of the AN/UYK-20 tactical data
3 2768 001 91550 7
DUDLEY KNOX LIBRARY